RAPTR: ROBUST ARTICULATED POINT-SET TRACKING

A Dissertation
Presented to
The Academic Faculty

By

Miguel M. Serrano

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2018

RAPTR: ROBUST ARTICULATED POINT-SET TRACKING

Approved by:

Dr. Patricio Vela
School of Electrical Engineering
Georgia Institute of Technology

Dr. Ayanna Howard
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Yu-ping Chen
Byrdine F. Lewis School of Nursing
and Health Professions
Georgia State University

Dr. Anthony Yezzi
School of Electrical Engineering
Georgia Institute of Technology

Dr. Matthieu Bloch
School of Electrical Engineering
Georgia Institute of Technology

Date Approved: August 20, 2018

I hated every minute of training, but I said, 'Don't quit. Suffer now and live the rest of your life as a champion.'

Muhammad Ali

In dedication to my loving fiance, caring family, reliable friends and all the role models that always supported me in my life.

ACKNOWLEDGEMENTS

This work is a collection of the various studies I have pursued in my time at Georgia Tech. From controls to robotics to computer vision to machine learning and all the topics in between, I'd like to send a special thanks to Dr. Vela for always supporting my curiosity, providing solid guidance when needed and whom never stopped challenging me.

In the IVA Lab, I thank my peers for always wanting to share new ideas. This goes out to Alex, Fujen, Yipu, Luisa and the rest. There's just too many.

I'd also like to thank my students, especially the few who collaborated with me to make these findings possible. In the most recent incarnation of my undergrad group, I'm referring to Ashar and Micheal.

To Dr. Howard and Dr. Chen, I'm forever grateful for the access to the world of physical therapy you made available and the guidance on my how my studies could have a positive impact.

Finally, I would like to acknowledge with gratitude, the support my family: My parents Juan and Jessie, my siblings Maria and Adrian, my extended family Armando and Oscar, my friends Hassan, Gbolabo, Kelvin, Mike and Mike and my fiancée Angela. They have been my bedrock and source for inspiration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this work is to present the Robust Articulated Point-set Tracking (RAPTr) system. It works by synthesizing components from articulated model-based and machine learning methods in a framework for pose estimation. Purely machine learning based pose estimation methods are robust to image artifacts. However, they require large annotated datasets. On the other hand, articulated model-based methods can emulate an infinite number of poses while respecting the subject's geometry but are susceptible to local minima, as they are sensitive to the various artifacts that appear in realistic imaging conditions (e.g. subtle background noise due to shadows or movements). The proposed work outlines how to drive the dataset generation using the same models employed in the model fitting to create a representative training set and how to include the trained detector's response in the model fitting strategy to introduce a robustness to artifacts and an increase to the solution's region of attraction. Furthermore, the articulated model serves as a shape and moment-based feature generator. A linear regression model trained on these features predicts the final pose estimate. When necessary, an intermediate representation is defined so that the two approaches may operate on compatible inputs. The proposed solution will be applied to articulated pose estimation problems where pose estimation accuracy is the priority.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Recently Machine Learning has become a prerequisite for most Computer Vision tasks. Advances in the field have had a profound impact on the computer vision community, making detection methods more reliable in complex conditions [1]. This is due in large part, to their ability to capture variations in the subject's appearance. These algorithms gain such a robustness to appearance changes by being trained on numerous samples. However, this can also be a hindrance. As the dataset is only a finite collection of appearances whose creation requires a large effort of collection and annotations to make it viable for use, the system's utility is heavily dependent on the user's ability to create a dataset which is both balanced and sufficiently large. Model-based methods can be included in the machine learning pipeline to bypass this limitation. Furthermore, these models can serve as feature generators, and provide additional information about the subject's pose when fitted to the capture data. The objective of this research is to provide a better means to exploit the benefits articulated model-based methods can provide a machine learning pipeline, and how the inclusion of the pipeline's prediction in the model-based fitting method can be mutually beneficial and ultimately return better pose estimates.

Advances in machine learning algorithms have made detection and classification solutions more robust, capable of identifying subjects of interest in uncontrolled and unmodeled environments. Furthermore, the rapid improvement in both the dedicated memory and computational capacity of GPU's has allowed for such algorithms to generalize better, as they can process larger sets of training data in a shorter period. In Computer Vision, this is a must. For example, when attempting to solve the pose estimation problem, any solution must be able to estimate the pose of subjects of different appearances. This is only natural as the subject can be of any sex, race or shape. Similarly, the algorithm should be

able to compensate for any clothes the subject is wearing and the environment they are in.

Previously, Computer Vision relied on features engineered to be easily detectable under varying circumstances. Prior to the advancements of convolutional neural networks, other feature generators like SIFT features, edges, silhouettes and textures were used [2]. Articulated models and imaging methods were designed to mimic how these features would appear within the camera view. Objective functions that minimize the discrepancy between the modeled and actual view of these features were then utilized to update the model's shape and group components until the cost function was minimal. Theoretically this would imply that the model's shape estimated the subject's. However, this would ultimately turn out to be a local minimum, heavily influenced by any un-modeled noise or imaging artifacts. Outside of highly controlled environments and capture conditions, when working with classical computer vision features there is no real way to handle unmodeled responses whose appearance would confuse the algorithm. Unfortunately, this limited the use of such algorithms, forcing pose estimation solutions to be confined within controlled spaces or specific capture conditions which were easier to model. It is for this very reason that computer vision began creating approaches that utilized as many features possible [3].

Now Machine Learning methods are incorporated in Computer Vision solutions to avoid such limitations. In the recent literature, there are three main types of approaches employed [4, 5]. The first is called 'Bottom Up', which applies a machine learning limb detector to identify one or many such limbs within a given image. This is followed by a model fitting approach, either through the fitting of a model via gradient descent [6] or a straight forward pose prediction [7]. The second approach involves a synthesis of the subject limb detections and features generated either from the predictions or directly from the view [8, 9]. The last approach is a direct pose prediction utilizing only the image as an input [10, 11, 12]. Within this work, only the two former methods are explored.

Limb detectors, as used in the 'bottom-up' approach, are trained on a vast array of instances that capture variations in appearance across different subject samples under dif-

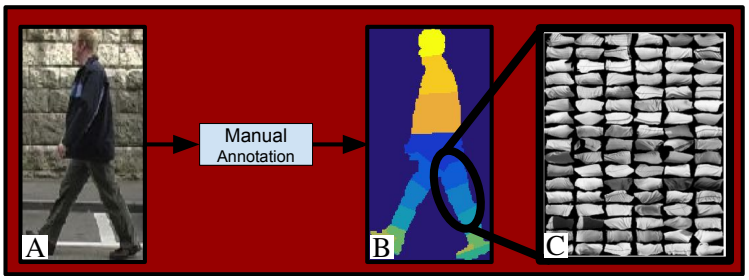

Figure 1.1: Dataset Generation: Every input image (A) is labeled by embedding a skeletal frame connecting the joint position annotations into different regions to represent the desired limb classes (B). Doing so for each image in the dataset, create multiple samples each limb (c).

ferent conditions, ensuring robust limb detection. Training datasets are made up of millions of annotations for a given limb, a requirement for any robust detector like a convolutional neural network [13] or a randomized decision forest [14]. Doing so allows them to account for the numerous variations in the subject's appearance from a single video, which can arise from motion, self-occlusion or simply from being seen from a different angle. One of the first examples is pictorial structures [15]. With only 9 features, the authors utilized a maximum likelihood (ML) estimator as a limb detector. As the years progressed, the limb detectors employed in pictorial structures were further improved by incorporating more advanced limb detectors [6, 10], more distinct features [16, 17], and more adaptive appearance models [3, 18, 19, 20].

Using limb detectors provides another advantage. If a limb detector is specifically designed for a given limb, when it processes an image the detector returns a likelihood map for the most probable region within the image containing its corresponding limb. Thus, if a single detector is trained for every limb on the subject's body, each detector would be applied to an image providing a series of likelihood maps denoting the areas each limb is likely to occupy [21]. Alternatively, whole body limb detectors have also been created to return the likelihood maps of all limbs [22]. Essentially, the detector acts as a filter that takes in an image, removes all the superfluous image information and returns only the signals needed for the pose fitting objective Figure 1.1.A-B . Thus, allowing for the

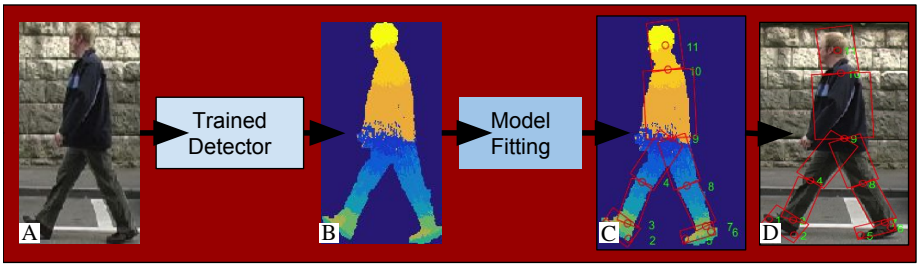

Figure 1.2: Model Fitting: The input image (A) is evaluated by the limb detectors to produce a limb likelihood map (B). An articulated model is then fitted to their response, producing an estimate for the subject's pose (C-D).

definition of the objective function to only account for the placement of the model's limbs to the subject's limbs as they appear in the projected view (image), free of any image clutter or artifacts. Also, as the solutions employ modeling methods that account for the subject's physiology and its projected view, the fitting problem becomes a limb matching problem. The limb detector's performance depends on the size and diversity of the dataset used for training, which means that the dataset must include various views and geometries of the same object type for it to be conducive to meet the training objective. A common practice is to collect an expansive dataset, using tools like LabelMe [23] to produce sets like ImageNet [13], which collectively have millions of labeled images for various objects. However, this approach is impractical as these datasets take years to collect.

Furthermore, these sets are exceptionally expensive. With a per annotation cost ranging from 1-7 cents and a viable dataset easily reaching a total of a million entries, an expansive set could cost anywhere from $\$10,000$ to $\$100,000$, making this a financial barrier to entry which can prevent prospective research purposes.

Additionally, limb detector datasets require more annotation than the standard datasets. Each section associated with a desired limb class needs to be manually segmented. Simply put, marking the limbs alone would be fifteen annotations. Adding the joints increases the set by eleven, and that is a highly conservative number simplifying the modeling of the entire torso as a single limb with a joint at the hip and neck. This means that the already heavy overhead to generating a good dataset for training, is worse when creating a full limb

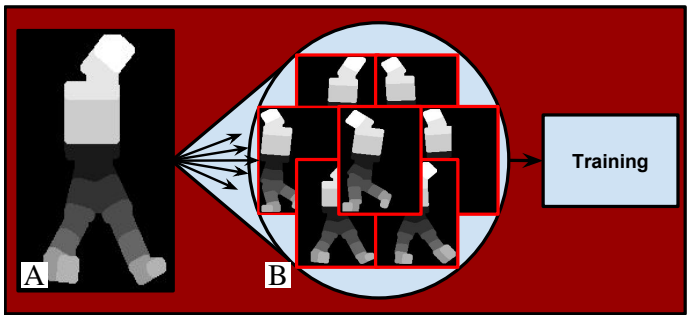

Figure 1.3: Illustrative example of an articulated model-driven sampling strategy: generates entire dataset from synthetic sample images rendered using the model.

detection set.

It should be noted that there are two types of datasets for human pose estimation. The first is the limb detector dataset. As previously stated, these sets include limb annotations within the image or 3D point-cloud that allow for the training of limb detectors, like the Human3.6 dataset [24] . They typically also have the joint positions as well. The second set only provides the images or point-clouds and the subject's ground truth joint positions. The latter is the most common dataset, implying that it is difficult to find many of the former type. Examples include [25, 26].

Fortunately, although limited, there are some publicly limb detector datasets available for the research community. The TUD pedestrian dataset [27] is one example that was created and manually annotated for training a limb detector. The process used to generate it is denoted in Figure 1.1.A-C. However, the shortcoming is that it only provides 400+ samples. Even with data augmentation, it is a rather limited dataset that only works within a specific use case of walking.

Recently, some methods have adopted the use of models in their dataset generation (1.3). In [28], mesh models are projected onto modeled views to generate sample range images. The samples span subjects of varying heights, weights, ages, and sex. However, the poses were captured using a motion capture system, limiting the range of possible poses to those specifically chosen for the initial capture. One publicly available dataset generated using a similar approach is the UBC [7]. Through the exploitation of the Makehuman

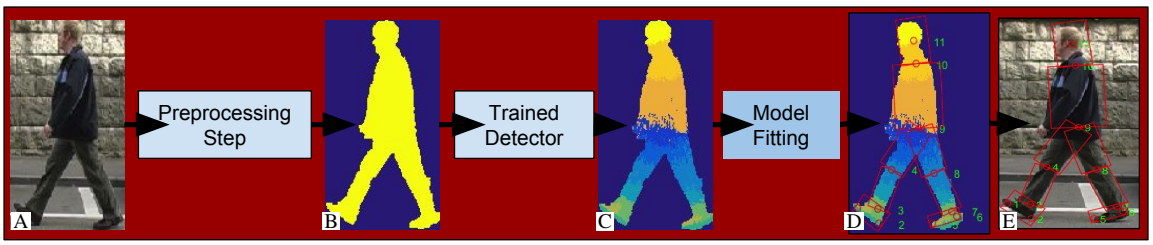

Figure 1.4: Model Fitting: The input image (A) is preprocessed to produce a feature map(B) that is evaluated by the limb detectors (C). An articulated model is then fitted to their response (D), producing an estimate for the subject's pose (E).

program [29], millions of samples of range image captures of humans with annotated limbs and ground truth joint positions are provided for public use. This dataset is used as a foundational training set is in this work.

Once the dataset is defined and the limb detector is trained and applied to the input image, pose estimation is achieved by optimizing an objective function (Figure 1.4). Given the output detections from the limb detectors, an objective function modeling a difference between the subject's captured state and the model's current state is defined. Essentially, through the optimization of said function, the model is fitted, with the resultant outcome being that the model ends up mimicking the subject's shape and thus estimating their pose. With respect to the modeling approach, given the limb likelihood map produced by the limb detector, the pose estimation is accomplished by fitting an articulated model so that the cumulative limb overlay response is maximal (Figure 1.3.A-E).

There are two types of modeling covered in this work. Modeling the subject is defined by either a loosely limbed model or a fully articulated model. For the loosely-limbed model, only the connections between the joints or limbs is defined. Like a graph, the joints are treated like vertexes and their connections like edges. Thus, if the centroids of each joint are detected, the pose estimate is then complete by assigning the connections. A fully articulated model, however, is defined by a collection of connected links whose geometry and degrees of freedom (DoF) mimic the subject's physiology.

One example of a loose-limbed approach is to locate the unique limb modes and, given

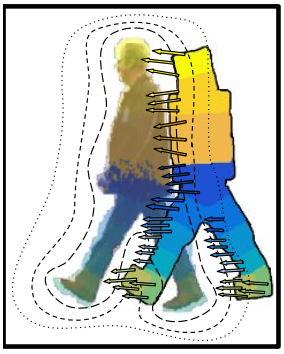

Figure 1.5: Depiction soft assignment model fitting behavior: contours and arrows represents the region of attraction and directed gradients, respectively.

the graphical dependence of the limbs, define the joint placement without strictly imposing link lengths [28]. Joint offsets are learned a priori and included in the final placement. Features can also be generated and fed into a machine learning regression algorithm like a linear model [7] or a Deep Convolutional Neural Network [8]. In these instances, a mapping from the set of features learned from the detected regions is mapped to the joint pose space.

Another approach is to formulate a fitting strategy with an explicit representation of the model, while imposing some constraints. As the model formulation is not differentiable, line search methods are applied [6] to estimate its pose. These solutions work by traversing the solution space with a semi-random strategy, approximating the gradient update by doing a directed search in the parameters space and sampling their error. The resultant update step is based on how the evaluated error changes with every sample. Unfortunately, this approach does not always produce consistent pose estimates. To reduce inconsistent pose estimates, model parameter limits are introduced into the estimator [30, 31, 32, 33]. These constraints reduce the pose recovery system's susceptibility to returning erroneous poses by incorporating the natural limitations or trained relationships of the subject's degrees of freedom (DoF).

Randomized search strategies were also developed. Simulated annealing, particle swarm optimization, particle filtering and their combinations were employed for the purpose of

pose estimation [34, 35, 36]. Using a similar strategy as the line-search methods, these algorithms evaluate a number of solution candidates with their respective optimization loss function, with each value serving as a sample of the solution space. Once an approximation of the solution space is achieved, these methods often selected the best fit answer or took the weighted average of all the tested solutions. Here, the weight is defined by the effective loss value, with the lower value sample getting assigned a higher weight. Particle filtering, uses a slightly different strategy in that it exploits the tracking kinematics to instantiate the search.

Fitting articulated models to the output limb detections is another approach to the pose estimation problem. As mentioned earlier, the whole line of pictorial structures [15, 3] employs this methodology. Robust point-set registration, was originally created for mesh point-set fitting problems, was use to fit skeletal maps to detected limb centroids [37]. However, in [38], it was extended to work as a method to register an articulated point-set to a point cloud model for the purpose of pose estimation.

In this work the RPSR algorithm is further extended to create the Robust Articulated Point-set Tracking (RAPTr) system. The RAPTr system's definition allows for the inclusion of a direct mapping between the model limbs and the limb detector's detections, demonstrating a direct connection between a model and machine learning approach. Furthermore, a claim of this work is that as the functional is density-based, it is differentiable. Its gradient can be taken, which will result in more direct convergence, leading to faster and more consistent pose estimation (Figure 1.5). Also, the density based formulation allows for the pose to be recovered given large displacements due to an extended region of attraction. Each advantage is a direct result of the merger between the machine learning and model fitting. In this work, three key contributions are presented to demonstrate the utility in combining articulated model and machine learning based methods. The first contribution of this work is how the articulated models can be employed to generate representative datasets. Each set is made up of the input images and label maps corresponding to sampled

poses undergone by the model. Doing so removes the collection and annotation overhead while ensuring adequate subject representation. Furthermore, as the articulated model defines the label maps, it therefore creates the correspondences between the model links and the detector responses. Also, because it is all automatically generated, the large barrier to entry in the pose estimation space is bypassed. The second contribution of this work is to develop a formulation that incorporates the limb detection correspondences in the model fitting. This is accomplished by incorporating the prediction weights in the gradient update equations, effectively affording RPSR the behavior of the iterative closest point (ICP) algorithm with the weights serving as a "soft assignment." Doing so introduces the extended region of attraction provided by RPSR, allowing for the system to estimate the subject's pose even if they or their limbs undergo large displacements. Although, this was partially accomplished by the pictorial structures solutions, a density based approach like RPSR is differentiable and produces the aforementioned advantages. The last contribution is using the feature generated from the pose estimate provided by the articulated model to train a machine learning regression function to return a final pose estimate.

1.1 Summary of Contributions and Outline

The thesis is comprised of four main projects, with the first two serving as components for the RAPTr system and the last two demonstrating its use in two pose estimation problems. The first application defines the limb detector definition. The second establishes a general pose estimation solution through the derivation of Robust Point-set Registration (RPSR) algorithm towards its use in an articulated model with the definition of "soft assignment" coefficients. The third applies RAPTr, tying the limb detection outputs to the model fitting approach in an intuitive way. The last application generalizes the findings and demonstrates how the RAPTr strategy can be extended to serve as a feature generator which can enrich the already established features and aid machine learning algorithms, returning better estimates with lower error.

Ultimately, the RAPTr framework can be included in any machine learning based pose estimation solution. Although, the machine learning was included to enhance the model fitting in this work, the converse is true. As the system's aim is to extract additional information about the subject's geometry by actually fitting an articulated model, any features generated from said fit naturally supplement the machine learning solutions output. This is depicted in Figures 1.6.A and 1.6.B for the training and prediction, respectively. The proposed framework's contributions are denoted in red while the classical machine learning components are represented in blue. Serving as an auxiliary function by splitting the pose estimation into a limb detection and joint position regressor, the additional layer of logic extracts further information that results in better pose fits. This will be explored in further detail in later chapters.

In Chapter 2 a limb detector is formulated and used to estimate clinical gait metrics from video collected of walking subjects. To ensure the detector is able to recognize the various components of a subject, the training dataset was enhanced by partitioning the area into anatomical regions using a skeletal model connecting the joint annotations. These region serve to model different sections of a walking subject's limbs. The work is then directed to locating the stance foot placements based on the integral image from the foot detections in an effort to extract the foot centroids and other desired gait metrics. A ground truth analysis of the system's performance is presented, demonstrating its viability to serve as consistent limb detector and clinical gait estimator.

In Chapter 3, a Mixtures of Gaussians based pose estimator is derived and applied to infant single leg pose estimation. Using the work presented in [37] for non-rigid registration as a foundation, a general reformulation for articulated rigid body pose estimation is presented. The method is then tailored for estimating the pose of an infant's leg. Furthermore, by incorporating the geodesic measure along the model and subject's surface, the preliminary formulation for defining a correspondence between different areas on the model and their corresponding regions on the subject is defined. Both quantitative and qualitative evi-

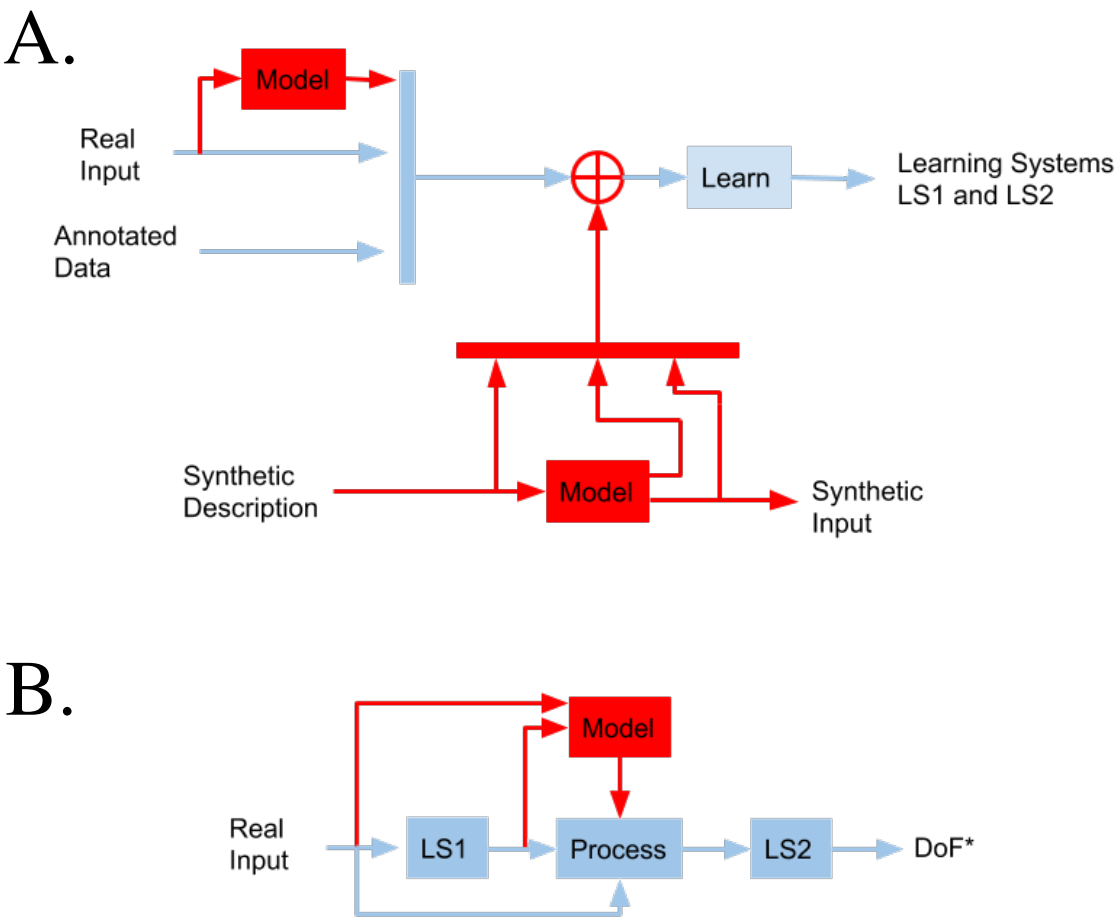

Figure 1.6: High level flow-chart of the proposed RAPTr Framework. Both the training module (A) and the prediction module (B) have the contributions from the proposed work colored in red with the traditional compenents shown in blue.

dence demonstrating its ability to estimate the joint trajectories for clinical evaluation, even when tracking fast and varying motions like spontaneous kicking, are presented.

In Chapter 4, the mixtures of Gaussians based pose estimator is extended to define the RAPTr system and applied to estimate the full-body pose of an infant captured from a single depth camera. Through the limb detector's predictions from Chapter 2 when trained on a synthetically generated set of infant images, a dramatic increase in the RPSR algorithm's utility is created. As RPSR is density based, it naturally allows for the synthesis of its formulation with the confidences provided by limb detectors to produce the RAPTr system. Additional benefits include an increase in its region of attraction, a robustness to point cloud

noise, an improved tolerance to false positive classification and an increased utility allowing for its extension to full 3d infant pose estimation. Both quantitative and qualitative evidence evaluating the system's performance on a real infant and a robotic infant are presented.

In Chapter 5, the RAPTr system is applied to estimate the full-body pose of humans, captured from multiple range image views. The formulation from Chapter 4 is versatile and capable of doing pose estimation for various subject types. In this application, the subjects of choice are adults. In this use case, the articulated model serves as both a moment-based and shape-based descriptor generator which, when concatenated with other, standard moment-based features derived from the limb detections provides improved 3d pose estimate errors.

Finally, Chapter 6 is the conclusion. A summary of the work and its contributions are outlined there, including a discussion of possible directions future work may take to extend this dissertation's findings.

CHAPTER 2

2D LIMB DETECTOR: AN APPLICATION FOR CLINICAL GAIT ANALYSIS

2.1 Introduction

Clinical gait metrics are temporal and spatial parameters that, when accurately measured, provide useful diagnostic information. In clinical settings, they serve to monitor the functionality of a subject's gait, and to provide a measure of performance for diagnosis and evaluation of the subject's progress under a therapeutic routine [39]. However, there is often a trade-off between the gait evaluation tool's accuracy and its versatility. Typical systems require advanced user expertise or much overhead in terms of setup and dedicated space [40]. Furthermore, these systems are often not present in less specialized wards. The aim of this work is to design a gait analysis system featuring minimal operational requirements and accurate estimation of clinical gait parameters.

Beyond minimizing overhead and maintaining accuracy, additional criteria that promote adoption need to be considered: a gait analysis system should be affordable, non-intrusive, easy to use and capable for at-home and outdoor environments. Making the system robust enough for outdoor assessment admits gait health evaluation in real-world scenarios like walking on grass or concrete. Having the system be easy to use, allows for a non-expert like a family member to do the collection to be later evaluated by the clinician when they are unable to visit the patient.

Automated clinical gait assessment tools have the potential to impact many communities. The ability to move one's self and maintain certain body positions is required for having an independent lifestyle [41]. Through the use of an automated clinical gait assessment system, objective evaluations of a patient's gait can serve to monitor their gait health. In particular it would greatly benefit the key demographics in need of frequent gait health

evaluations. This is true for cases of head trauma, where degeneration of the gait functionality can be a symptom [42] prompting the need for therapeutic intervention. Also in case of the elderly, the metrics the system provides can serve as viable fall predictors [43] or to help explore and understand the underlying mechanisms of aging affecting the patient [44]. For patients with Alzheimer's [45] or dementia [46], such a system can help track the progression of their impairment. With respect to patients with newly obtained prosthetics [47], such a system would serve to evaluate their progress towards relearning how to walk. In all the aforementioned cases, the system could serve to help therapists in identifying the effectiveness of their therapeutic routine through objective evaluations of the patient's progress.

Gait health evaluation most often involves observing the patient walking, which is an action whose major joint articulations occur in the sagittal plane. Therefore, a single view suffices for application of traditional observational gait analysis techniques [48]. Thus, although multi camera arrangements can provide 3D gait information, monocular camera setups have the potential to provide sufficient gait measures for gait health evaluation. The natural next step is to develop an automated computer vision system capable of detecting the same cues a therapist would when estimating important clinical gait metrics.

This chapter presents a method to estimate the following clinical gait metrics from video: stride and step length, walking speed, cadence, heel strike and toe-off angles. These measurements are sufficient to predict a subject's condition relative to known population norms [39]. The method employs a robust foot detector that is not subject specific. Its invariance to background noise allows for the processing of video collected using consumer cameras, making it possible for clinicians and non-specialists alike to capture in at-home and outdoor environments.

2.2 Related Works

Although motion capture systems are capable of estimating to high accuracy the gait kinematics of walking [40], the infrastructure costs and setup requirements constrain where, how, and by whom gait kinematics may be collected. Furthermore, only a limited interpretation of the gait is possible since subjects are often recorded on treadmills; the gait kinematics of walking on a treadmill differ from those of walking on a solid natural surface [49]. Contemporary research seeks to relax these constraints and democratize the data collection process by requiring only a single camera device to capture the subject's walk in unconstrained natural environments, then applying advanced computer vision techniques to automatically analyze the subject's gait health. A further benefit would be more efficient clinical analysis by experts, since the preliminary analysis is done automatically.

The recent commercial success of depth sensors, such as Microsoft's Kinect camera, have led to investigations into their clinical use [50]. Depth sensors provide RGB-D data decomposed into a standard color image and a separate depth image. The Kinect SDK provides software to estimate articulated 3D poses from depth images of a human subject. Contemporary commercially available depth sensors are designed to operate indoors only (outdoor-capable range sensors cost an order of magnitude more, at minimum). For indoor setups, the reported accuracy supports clinical analysis [51].

Validation of the Kinect's feasibility for clinical assessment has been shown for a subject walking on a treadmill [50] or across a room [52]. The former places the Kinect in front of the patient as they walk on a treadmill, then uses the Kinect SDK to get the joint angles versus time for processing, to then recovering the gait metrics. The latter investigation sets the Kinect on a perch in the corner of a room, looking down at the subject. Exploiting the planar floor surface to calibrate the Kinect, the foot placements are recovered by thresholding the captured 3D point cloud by the height above the floor plane.

Capturing longer periods of walking requires moving the camera as the subject walks.

One way to do so includes instrumenting a walker with a depth sensor [53]. Given that the sensor placement for this setup is too close to capture the full body, the articulated model to fit consists of only the legs. Alternatively, the task of subject tracking may be automated by a mobile robot. In [54], tags placed on the subject's feet and lower torso aided the tracking and gait analysis. Both instances demonstrate an effort to monitor gait activity in less constrained indoor environments.

Since monocular vision solutions using color cameras lack the depth information associated to RGB-D cameras, additional processing or constraints are required to address visual clutter. Constraints include a highly controlled environment whose visual background is highly uniform, as well as requiring specific garments or trackable markers to be worn [55] or as little clothing as possible [56]. Alternatively, subject specific appearance template models have been shown to work [57, 58]. The templates are non-adaptive models that assume the subject's appearance will remain consistent during data collection, effectively using the subject's appearance as a substitute for markers. Model-based methods have been found to have strong results fitting subject-specific articulated models from video [59, 60], but their clinical utility has not been validated. To relax the constraints, heuristics may be applied in order to limit the solution space and thereby prevent unrealistic kinematic estimates. As an example, statistical anatomical body proportions have been used to locate the desired subject limb joints given the detection of key body points of reference, like the neck or hip [61, 62]. To work, a subject's body proportions should fall within the tuned model's parameter range.

2.3 Methodology

This section describes the proposed procedure for automatically extracting clinically-relevant gait metrics from a video sequence (Figure 2.1). First the subject's walk is captured by following the prescribed capture protocol. Second, the video is processed to extract the subject from the image by removing from consideration all pixels associated with the background,

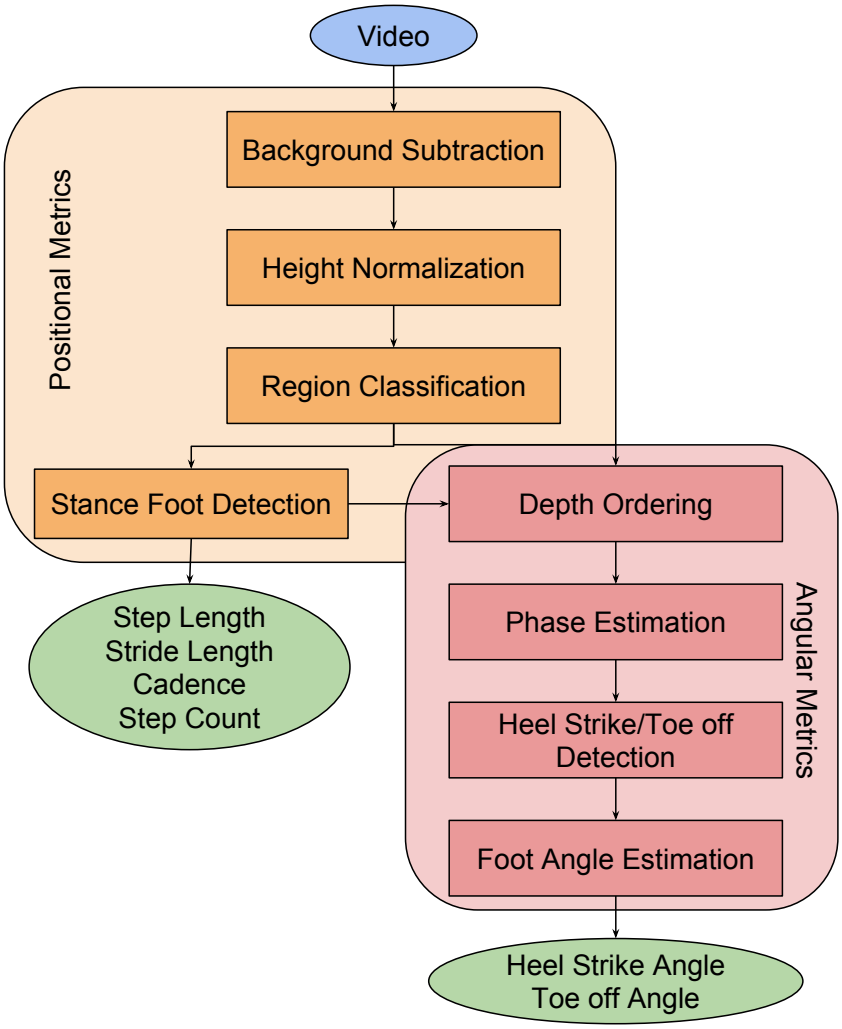

Figure 2.1: A flow chart outlining the logical order and dependence of each step used to estimate the clinical gait metrics.

leading to a silhouette image of the subject. The subject's silhouette is then resized to meet the required height specifications of the trained limb detector. At this point, the main processing components come into play.

The height-normalized silhouette serves as input to a body parts detector. Specially trained classifiers are applied to the silhouette image, leading to pixel-wise classifications of the image. Classification categories include the sole, the heel, and the top part of the foot, as well as other body parts. The primary pixel-wise classification method is the randomized decision forest (RDF) [14]. RDFs were chosen for classification because of their robustness to noise. An RDF is an effective classifier, even in the presence of artifacts in the data. A forest, here, consists of a collection of decision trees, each of which carves out regions in the data space through a conditional sequence of binary tests. The outcomes of the binary tests over the training data successively discriminate between the different class regions within a given silhouette image. In the case of classification, the carved regions at leaf nodes are modeled by a normalized histogram denoting the likelihood of an object class terminating at that lead.

Lastly, the portions of the silhouette labeled as foot class are then sent into two paths. The first path identifies their stance foot placement and returns the stride and step length, step count, and walking speed. The second path uses the foot detections and the stance foot placement to identify the step's depth ordering and solves for the heel strike and toe off angles. Each of these steps is described in further detail in the remainder of the document.

2.3.1 Capture Protocol

Physical setup of the system involves the placement of a digital camera on a static support surface whose viewpoint is orthogonal to the walking direction of the subject, and captures the linear path of the subject. The camera can be any type that supports digital video recording (e.g., a digital consumer camera, a webcam, or a mobile device camera). Typically available support surfaces include a tripod or tabletop. The background scene is

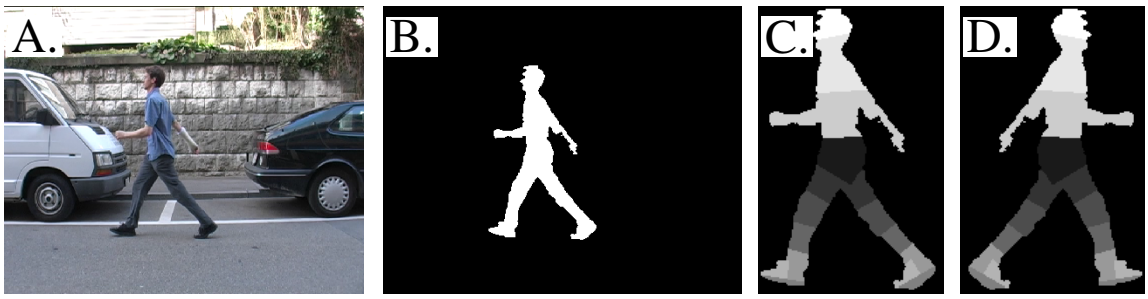

Figure 2.2: Depiction of an input image (A) together with the expected output (B) of the background subtraction step. The gray scale ground-truth output of the body labels are depicted in (C) with the mirror image in (D).

presumed to be predominantly static.

2.3.2 Subject Detection and Silhouette Extraction

A mixtures model based method is used for the background subtraction. Given a static camera viewing a mostly static scene, mixtures of Gaussians background modeling algorithms are capable of detecting objects that enter and cross the scene with high reliability [63], while being robust to the slow changes in both subject and background appearance.

Differences in the input image relative to the background model yield the desired foreground estimate of the subject, as a silhouette image (see Figure 2.2). A hard threshold on the likelihood of a given pixel not coming from a pixel's underlying intensity distribution is used to discern between foreground (subject) and background (the environment). Various groups of pixels are produced using this approach. Acceptable range limits on the silhouette area uniquely detects the subject's silhouette versus spurious (small) foreground detections due to image noise.

2.3.3 Height Normalization

Since the training process involved normalizing the pixel-wise evaluations versus a canonical target height, deployment of the classifier requires height normalization to be performed on the video sequence. This is true for both the RDF training and spatial limb likelihood.

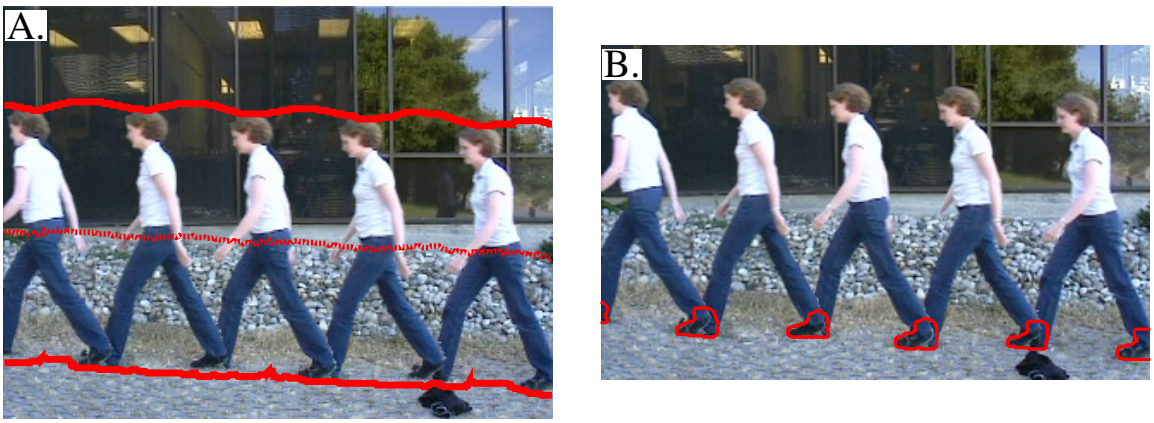

Figure 2.3: The input image is pre-processed to estimate the subject's height (A), then height-normalized for foot detection (B).

From an initial pass of the background modeling algorithm, the detected subject's silhouette size is calculated from the processed image sequence statistics. The images are then resized to match the training set height. Doing so compensates for the height variability across subjects, by providing as input to the RDF silhouette imagery compatible with the training data. An illustrative example demonstrating the upper and lower detections of the subject including their effective mid-line is presented in Figure 2.3.

2.3.4 Region Classification

Generating the Training Set

Prior to deployment, a random decision forest first needs to be trained from a representative dataset. Here, the annotated samples in the dataset provides evidence for silhouette images likely to be witnessed, thus the dataset should cover the range of expected observations. Doing so aids the model in accounting for the various poses taken during walking. To facilitate the training step, we generated the annotated samples from a pre-existing corpus, the TUD pedestrian dataset [27]. The dataset consists of manually annotated silhouettes for the respective joints of the subject, together with annotations of the subject's head, neck, hip, knees, toes, heels and ankles. These base annotations are further processed to create

the regions of interest within the silhouettes that serve to train a limb detector RDF. An example of the data, silhouette and class regions is presented in Figure 2.2. A mirrored version of each training image is added to account for walking in the opposite direction.

The additional processing involves further decomposing the body into ten labels (beyond the seven given in the TUD dataset). The additional labels highlight the key regions of interest associated with gait analysis. For example, the feet are broken up into three regions: the sole, the hind foot, and front of the foot. The additional labels improve the specificity of foot detection, thereby reducing the class confusion associated with ambiguous silhouettes.

The torso and head labels from the pedestrian dataset annotations are preserved. For this application, the hands and arms of the subject are not specifically labeled. Arm regions are labeled depending on their relative height with respect to the hip, torso, and head labels. Region growing is use to assign labels to arm regions below the hip line to prevent mislabeling them as thigh regions.

Randomized Decision Forest for Foot Detection

A randomized decision forest consists of T decision trees. The T decision trees in an RDF classifier return a strong classification through a series of responses from weak classifiers. Each tree is made up of split and leaf nodes with n and l indexing a given split and leaf node, respectively. At each split node, a decision is made using a "weak learner" with the feature parameter $\theta = (\phi, \tau)$, where $\phi = (u, v)$ denotes the relative pixel locations to query and τ is a scalar threshold associated to the binary decision. To make a prediction for pixel x, the algorithm begins at the root node of the tree and travels down according to the (binary) decision function response at each node visited. For a node n with parameter θ_n the decision function is defined:

$$h(x; \theta_n, J) = f(x; \phi_n, J) \geq \tau_n, \tag{2.1}$$

where a false outcome branches to the left child and a true outcome branches to the right child. The feature function $f(x;\phi)$ performs local pixel comparisons given a scalar-valued input image, here the silhouette image J,

$$f_\theta(J,x) = f(x;\phi,J) = J\left(x+u\right) - J\left(x+v\right), \tag{2.2}$$

where u and v are 2D pixel offset vectors. Feature evaluation repeats until a leaf node l is reached.

Training of a decision tree involves identifying the node test parameters θ_n. The offset parameters (u,v) are sampled randomly from within a bounding box. For consistent sampling across the training samples, the bounding box was determined by the average height of the training silhouettes. Setting the bounding box radius to a quarter of the average height returned the most favorable results. Furthermore, the training process involves setting v to $[0,0]$ half the time, to capture the univariate information for the given pixel. Each parameter is tested by evaluating Equation 2.1 on the training data used to define the node. The parameters that return the largest information gain are then selected.

Selection of the τ parameter is chosen from a finite set of values. Given that a silhouette is a binary mask of the subject, a fair assumption to make is that any comparison would lead to a feature response of $\{-.5, 0, .5\}$. Thus for every randomly sampled ϕ, each possible value of τ is tested.

To ensure the discrimination power of a tree, the feature selection process involves maximizing the information gain of the binary decisions. For a given tree, each node is defined according to the following algorithm:

1. Randomly propose a set of candidate ϕs.
2. Partition the sample training dataset $Q = \{(c,J,x)\}$ into left and right subsets based

on the outcomes of the binary test

$$Q_l(\phi) = \{(J, x) | f_\theta(J, x) < \tau\} \tag{2.3}$$

$$Q_r(\phi) = Q / Q_l(\phi) \tag{2.4}$$

for each possible value of τ.

3. Keep the $\theta == (\phi, \tau)$ that returns the largest gain in information:

$$\theta^* = \arg\max_{\phi} G(\theta) \tag{2.5}$$

$$G(\theta) = H(Q) - \sum_{s \in l,r} \frac{|Q_s(\theta)|}{|Q|} H(Q_s(\theta)) \tag{2.6}$$

where H is the Shannon entropy computed on the normalized histogram of body part class labels $c(J, x)$ for all $(J, x) \in Q$.

4. If the largest gain $G(\theta^*)$ is meets a predetermined acceptance criteria, and the tree depth is below a maximum, then recurse the left and right subsets $Q_l(\theta^*)$ and $Q_r(\theta^*)$.

The acceptance criteria is defined prior to starting the tree. Possible reason's for ending the tree's continued expansion are the node's population, the current depth, or that at that leaf the candidate θ returns a nominal information gain value.

The training dataset for each tree utilizes bagging, whereby the dataset is generated from a randomly sampled subset of training images and ground truth. Bagging ensures that each tree can account for different trends in the data while not over fitting to the entire training set. The Technique enhances the classifier's generality and improves classification performance for novel data inputs.

Classification is performed pixel-wise on the silhouette image J, resulting in a set of leaf nodes reached $L(x) = \{l_t(x)\}_{t=1}^{T}$ for the set of trees. At each leaf node the learned class distribution $P_t(c|l_t)$ for class c is obtained, with the pixel's final distribution defined

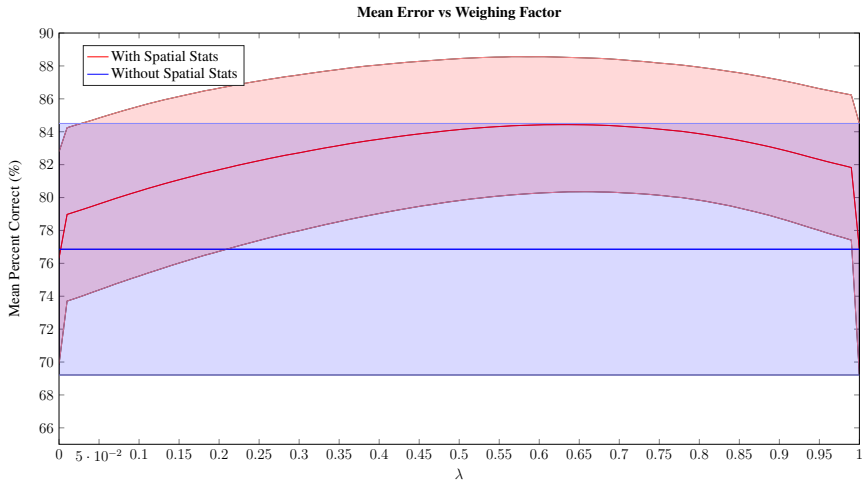

Figure 2.4: Analysis of percent correct across all classes as a function of λ.

by the average response in the set L according to

$$P(c|L) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|l_t). \tag{2.7}$$

Spatial Statistic for Improved Classification

The RDF provides a localized classification strategy for decomposing the silhouette, which is important for achieving translation invariant classification. However, it does not take into account more global trends in the overall training dataset. As the RDF provides per pixel classification based on the local geometry, it can often return erroneous labels (e.g. labeling pixels at the head as feet and vice-versa). We propose to incorporate also a more global, spatial classification probability, that computes each label's probability of occupying a pixel, given its placement with respect to the silhouette's center. The purpose of the label statistic $P(C|x)$ is to reduce the occurrences of the false labeling (e.g., it should improve the false positive rate, but should not impact the false negative rate).

There are two main instances of misclassification which can hinder the gait analysis. The first and most apparent are false negative foot detections near the feet. These can result in deformations in the detected foot regions and as result can cause the effect foot centroid

to shift from the desire foot center. The second is a false positive foot detection near the mid to upper body. Although these detections are often negligible because the majority of the detections reside on the actual feet, they can still be detrimental to the depth ordering analysis that comes after the stance foot detection.

Treating each annotated training sample as a centered example of a subject in the midst of a walking gait pose, the spatial statistics are calculated. Assuming the subject is centered, each pixel's likelihood is calculated based on the pixel's coordinate with respect to the expectation of the subject's silhouette. Doing so ensures a consistent strategy to set a reference frame for the spatial statistics of a subject in a normal image. The actual likelihoods are calculated by creating a stack for each class and taking the mean along the stack.

Assuming that the probabilities are independent, they are best combined in a multiplicative space. In particular, the final label probability is arrived at from a weighted geometric average:

$$P(c|J) = P(C|x)^{1-\lambda}(\frac{1}{T}\sum_{t=1}^{T} P_t(c|l_t))^{\lambda}. \tag{2.8}$$

with $0 \leq \lambda \leq 1$. A parameter sweep over λ establishes the optimal weighting, which improves the classification results. Figure 2.4 demonstrates that the optimal λ in terms of classification error for this application is roughly $.64$ resulting in an 84.5% error rate which is an improvement of 7.64% over the RDF-only solution. It is evident that most classes improve with a few incurring an error. This trade off however is justifiable as will be apparent when the 3 foot subclasses are used to estimate the heel strike angle.

2.3.5 Stance Foot Detection and Basic Gait Analysis

The RDF, when applied to the silhouette images, returns an estimate of the likelihood of a pixel belonging to a particular body-part class. Each pixel is assigned a unique class label that is defined by the class with the maximal probability, as per equation (2.7). The class probabilities associated with the feet are monitored over time. Taking their integral with

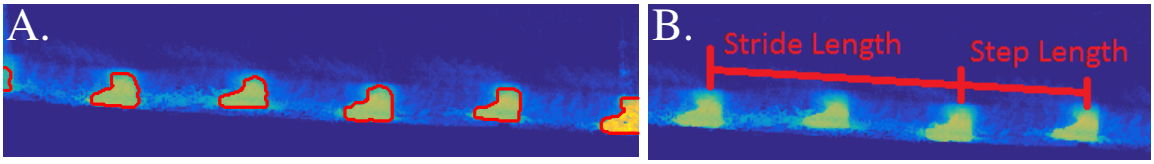

Figure 2.5: Integrated likelihood output of the random decision forest associated to the feet labels (A), Illustration of the measurement strategy employed to measure the step and stride length (B).

respect to time at each pixel provides a density map denoting the probability of stance foot having occupied a pixel. This density map leads to the estimate the static foot placement.

The period in the gait where a single foot is placed on the ground to propel the subject forward is known as the stance phase. The foot with ground contact is referred to as the stance foot, whose location over time provides all the information required to estimate the desired clinical gait metrics (step and stride length, cadence, and step count). Since stance feet have a relatively long static duration time, they are associated to pixel regions with high cumulative response relative to other parts of the image (Figure 2.5.A). Thresholding of the cumulative response image is used to extract the image regions associated with stance feet. We will refer to these as stance foot regions.

Estimating the stance foot position is accomplished by taking the centroids of detected stance foot regions. The averaging involved in computing the centroid has the effect of spatially filtering the foot position estimates.

The clinical gait metrics are then calculated from the stance foot position estimates. Step length is defined as the distance between each static foot placement as follows:

$$
\begin{aligned}
StrideL &= x_{k+1} - x_{k-1} \\
StepL &= x_k - x_{k-1} \\
Cadence &= K/L \\
StepCount &= K
\end{aligned}
$$

with x_k equal to the spatial location of the kth detection for all K detections and L the video's duration. As the stride length is the distance between each foot placement of the same foot, it is calculated as the distance between every other static foot placement (Figure 2.5.B). The cadence is defined as the number of detected static feet during the subject's walk. Lastly, the walking velocity is estimated from the ratio of the cumulative step length with the time interval over which the subject was detected.

2.3.6 Additional Gait Analysis

Foot Depth Ordering

Depth ordering, with respect to walking, is defined in this document as the identification of which foot is currently in its stance phase. Each stance foot is treated as a unique object. Thresholding the cumulative response of the feet classes leads to the stance foot region detection (Figure 2.6.A.1). With the regions isolated, a numerical value is assigned to uniquely identify them (B).

Using the regions as binary masks provides local areas of interest for establishing the depth ordering. Walking is a highly structured gait in which a single foot is always in contact with the floor. Thus at each instant in time during the swing phase, a single foot is traversing a single region of interest. During the swing phase period the local pixel values will fluctuate. The magnitude of the change depends on whether the foot crossing is the occluding foot or not.

First the periods during which any foot traverses a region of interest are estimated. This is accomplished by setting a threshold on the intensity changes at the detected foot placement point (Figure 2.6.A.2). Identifying the single vs double foot stance phase becomes a trivial outcome of this step (Figure 2.6.A.3). Summing across the signals and locating the periods in which two feet are in the stance phase at the same time results in the double foot stance period detection. Next, the distance of the current foot detections centroid with respect to each stance foot detection is used to identify the temporal order in which the

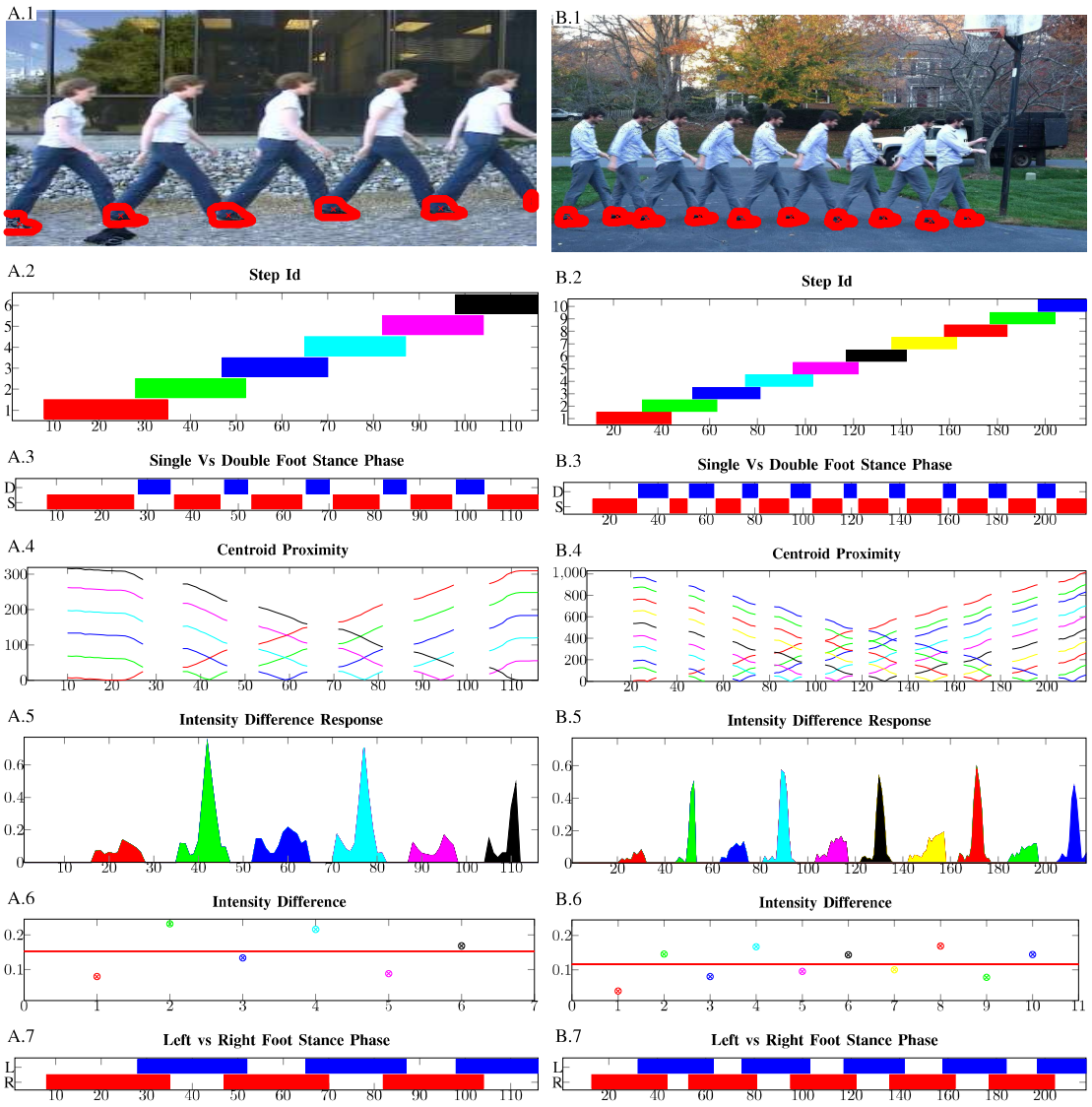

Figure 2.6: Demonstration of depth ordering solution on a normal gait (A) and atypical gait (B). Foot detection results (1) lead to the identification of the unique stance foot periods (2) and consequently the single and double foot stance periods (3). Using these results, the foot proximity signal (4) and the temporal pixel differences (5) in each region of interest leads to the definition of a differencing boundary (6) that classifies each foot as left or right foot (7).

Figure 2.7: Representative example of the foot orientation solution. Green denotes the upper foot, while read represent the bottom. The line with two end points shows the boundary estimated using the weighted SVM.

stance foot phases occur. In Figure 2.6.A.4 the foot detection expectation's distance with respect to each individual stance foot is presented. The resultant signals with the lowest magnitude denotes which stance foot region is currently in the stance phase.

With the stance foot placements ordered in time, the differencing magnitude determines the current swing foot's depth ordering. Taking the pixel intensity average magnitude change results in an adequate signal to differentiate between the two. The detected single foot stance phase periods are used as temporal windows for the expectation evaluation. In Figure 2.6.A.5 the effective average pixel intensity differences over time for the respective regions of interest are presented. Now determining the depth order can be solved using classification. Taking the expectation of these magnitudes creates a binary classifier whose boundary can distinguish between the two classes. An average value greater than the boundary implies the current stance foot is being occluded and opposite if less than the boundary (2.6.A.6). Identifying the left vs right foot becomes a simple task by including the direction of the subject's walk in the analysis Figure 2.6.A.7 .

Heel Strike and Toe Off Angle Estimation

An important metric of a gait's quality is the angle of contact with the floor during the beginning (heel strike) and end (toe-off) of the stance phase period. These moments are referred to as the heel strike and toe off, respectively. For the purpose of estimating the foot angle, the foot is broken up into three subregions. Each subregion represents a key landmark of the foot. These are the heel, toe and sole as denoted in Figure 2.2. These regions are included as classes during the training of the RDF and limb spatial likelihood

function. Defining the upper foot group as the union of the toe and heel class, and the lower foot group as the sole class creates two distinct groups, the upper and lower foot sets. Solving for the linear boundary between the two regions establishes an estimate of the foots orientation. This is essentially a hyper-plane problem. Support vector machines (SVM) are used for such problems. Treating the upper and lower foot regions as distinct classes and applying a Support Vector Machine (SVM) to solve for the boundary produces results in an estimate line whose slope proxies the foot angle measurement (Figure 2.7). A weighted variant is used to give priority to the upper foot classes. These angles are calculated during the beginning and end of the stance phase.

2.4 Results and Discussion

In this section, the application and evaluation of the pipeline to recorded video of walking subjects in diverse environments. Error analysis for the estimated clinical gait metrics is presented. First, an analysis of the system's performance for estimating the step and stride length, cadence and walking speed is presented. Next, the accuracy of the heel strike and toe is presented.

Data Acquisition and Ground Truth Generation

The proposed system provides a series of measurements, each of which requires their respective analysis. These measurements include the detected centers of the stance foot detections, the identification of the various phases of a gait, and the estimation of the heel strike and toe off angles.

To validate the system, various videos of subject's walking orthogonal to the camera's optical axis were used. The dataset is comprised of both publicly available and privately collected videos. For each subject, the camera was set at a fixed height and distance from the subject's path, and remained static during the capture duration. Sample videos of 14 different subjects from three outdoor sites and one indoor sites were used. The resolu-

Figure 2.9: Sample images of the evaluation video set.

tion of each capture varied. The collection was comprised of 32 videos, with at least 5 steps per video. Each video capture is used to evaluate the clinical gait metrics and the phase detections. However, only 5 videos are used to evaluate the heel strike and toe off measurements.

The experimental procedures involving human subjects described in this paper were approved by the Institutional Review Board.

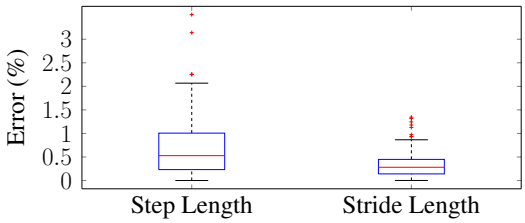

Figure 2.8: Percent error of estimated gait metrics.

Condition at each site varied. The outdoor settings demonstrated the standard difficulties associated with background appearance variation due to the weather. This includes the presence of shadows and glare created by the sun. Furthermore, subtle changes in the appearance happened because there were leaves rustling in the background due to the wind. Sample images of the outdoor locations are presented in Figure 2.9.A-C. The indoor scene presented different difficulties (Figure 2.9.D). The capture took place in a hall way. Throughout the walk, depending the subject's proximity to the back wall, shadows formed often confusing the background subtraction code and creating artifacts in the foreground estimates.

Ground truth for the sequences are established through manual annotation. There are

two types of measures made by the system. The first type corresponds to quantities that can be directly estimated from the images. These include the stance foot placements, the identification of the current phase of each foot, and the heel strike and toe off angles. The second type are the measurements that are a function of the first type. They include the stride and step length, cadence, step count, single vs double stance phase and the depth ordering (e.g. right vs left foot). Thus, two types of analysis were pursued to validate the system.

Manual annotations are used to define the ground truth for the first type of measures. For the foot centers, a user located the frame when the foot was completely planted onto the floor and selected its center. They did so for each identifiable stance foot occurrence. The annotator determined the phases by identifying the frames in which each foot underwent a change in their gait phase. Lastly, both the frames of the heel strike and toe off of each foot were selected. The corresponding angle of the foot was then estimated by manually selecting a point on the heel and toe of the foot. A line was then drawn between the two and the angle was measured.

The manual annotated ground truth was used to establish the ground truth of the second type of measure. Taking these values as inputs of the functions that return the clinical gait metrics served as ground truth for the step and stride length, cadence, and step count. Furthermore, the identified periods of transition were used to define the truth values for the single vs double foot stance phases, the depth ordering and the left vs right.

Error Analysis

Evaluating the system's performance using the detected stance foot centers returned favorable results. Analysis of the ground truth indicates that the centroid estimation error achieves sub-pixel accuracy on average, as indicated in the box-plot in Figure 2.10. Error is defined as the absolute difference in pixels between the estimated and ground truth foot center. The box-plot presented demonstrates the collective error across all videos in each

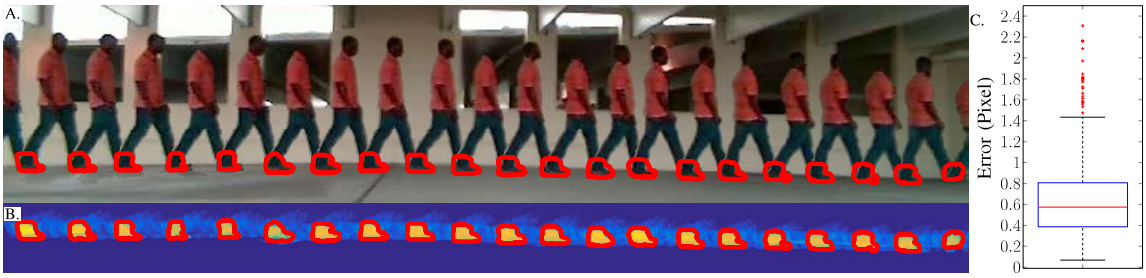

Figure 2.10: Visualization of the automated processing. (A) A composite of the detected subject with estimated stance foot regions overlaid on the horizontally cropped, height-normalized background image; (B) a cropped false-color image of the cumulative foot likelihoods with red/orange indicating high likelihood; and (C) the error (in pixels) of foot position estimation.

environment.

Analysis of the clinical gait metric estimates demonstrate that they can be reliably estimated. The corresponding percent error for the step and stride length is demonstrated to be within a 1% tolerance on average; see Figure 2.8. The cadence, walking speed, and the left vs right identification has 0% error as the system detected each step and correctly determined the depth ordering. Lastly, the transition error is on average 1-2 frames, implying that the estimates of single vs left and stance vs swing phase each foot are within 1 frames error as well. Also, the videos are over a hundred frames per capture, implying that the maximum error is always within 1%. Furthermore, the periods defined by the detected transitions are adequate for the heel strike and toe off calculations.

Although noisy, the angular measure of the heel strike and toe off angle are within an acceptable tolerance. The average angular error of the estimates is 3.19^o error and presented in Figure 2.11.C. A representative example of the process applied to a subjects walk and the estimated angles over time when applied to both feet is depicted in Figure 2.11.A-B. The major outliers in the angle estimates occurred as the subject was reaching the or traversing the boundaries of the image.

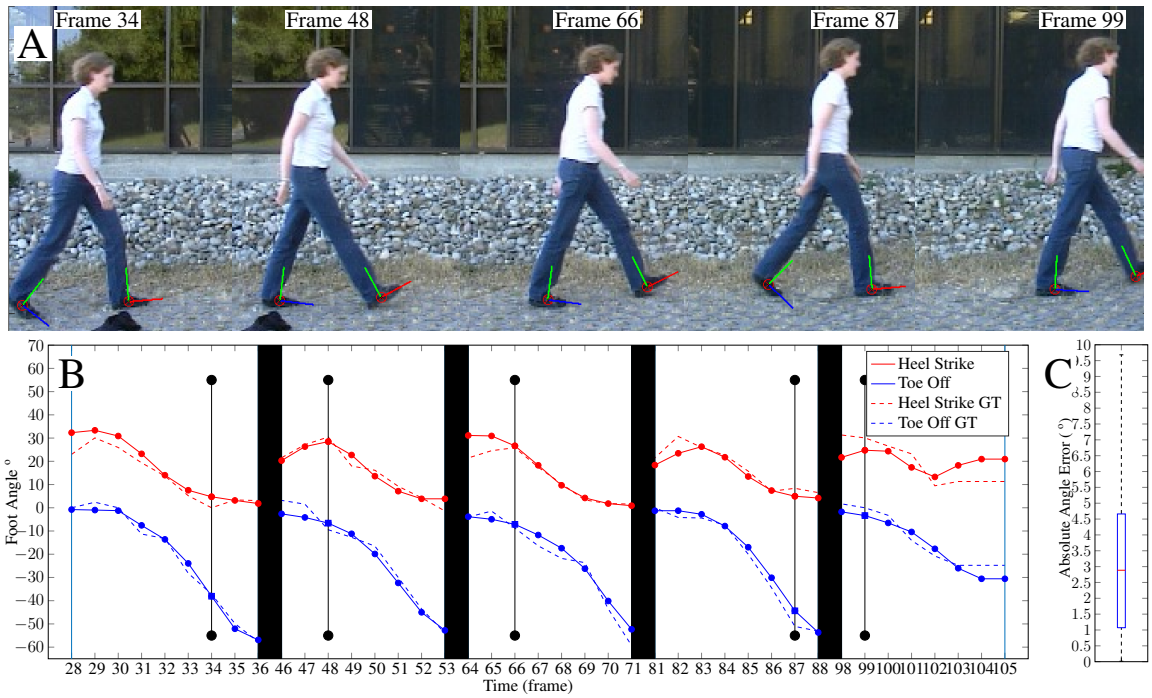

Figure 2.11: Foot angle during heel strike and toe off phases.

Discussion

Error analysis demonstrates that the approach is capable of estimating the desire metrics. For illustrative purposes, two sample runs are processed using the foot detection and depth ordering. The results of their evaluation using the proposed framework are presented in Figure 2.11.A-B. The samples include a subject with a perfect gait and another with an impaired gait. To replicate the impaired gait, the subject wore a cast of their leg. Each of the signal estimated using the methodology from Section 2.3.6 are included in Figures 2.11.A-B.2-7. The system is able to identify the moment of transition from stance to swing phase for each leg. Also, the determination of left vs right was correctly done using the estimated signals. For example, 2.11.A-B represents the change in pixel intensity used a measure to determine the current state of a given leg. A clear pattern emerges for a foot depending on whether it is the occluding or occluded foot. The result of such a decision leads to the correct detection of whether the foot is the left or right.

Measurements which are a function of the stance foot placement are the most reliable to measure. Values like the stride and step lengths are funtions of the stance foot center detections integral over time. This point is calculated from the integration of all foot detections of over time. As the stance foot is constantly detected and the mask integrated, the final integral image lead to a set of clearly distinguishable foot placements (Figure 2.10.B). By virtue of the integration, these measurements are robust to noise. This allows for the consistent measure of the basic gait metrics.

Using specific foot locations such as the front or back of the foot to denote its position is not advisable. While the background subtraction that provides the silhouette is robust to minor changes in the subject's appearance, every so often the gross shape of the subject's feet is not preserved, which impacts the classification outcome local to the feet. These minor inconsistencies introduce noise into the estimates, making specific foot landmarks unreliable.

The toe off and heel strike measurements, on the other hands, are highly sensitive to false positives. A weighted SVM is used to account for this susceptibility to missclassification. Furthermore, for these estimates the subject is required to walk closer to the camera. This is because, a higher resolution foot capture is required to consistently isolate the foot's shape. For this reason only 5 videos from the set could be processed for this analysis. The set of five included 3 captures from indoors and the two sample videos from Figure 2.6.A-B.

Accounting for the contrast between the subject' clothes and the test environment can lead to consistent results. The silhouette images over time are the primary inputs to the processing pipeline. On account of this, ensuring that there is a noticeable difference (for the algorithm) between the subject's clothing and the background scene will maximize the accuracy of the silhouette shape.

2.5 Conclusion

The motivation for this study was to create a clinical gait assessment system capable of estimating common clinical gait metrics while still meeting certain criteria. The criteria are that the system is affordable, easy to use, capable of working in non-controlled environments (e.g. outdoors or at-home) and that was still accurate enough to return consistent metrics. Such a system would create an immediate benefit to the medical community as it would allow for monitoring capabilities to be easily accessible. Furthermore, evaluation of a subject's gait health would not be restricted to the clinic. With such a method, clinicians will be able to assess the subject's walking performance in real-world environments like walking on concrete sidewalks or on grass.

The system detects and estimates the static placement of the subject's stance foot to compute key gait metrics used for predicting the health of a subject's gait. Additionally, it identifies the gait phases of each foot and returns estimate of the foots inclination at during the toe off and heel strike. The system is robust to image noise and presents an opportunity for at-home and outdoor gait evaluations using consumer cameras.

There are a few limitations to the system that can be the focus of future work. Shuffle steps or steps where the subject does not clear the stance foot with the swing foot during the swing phase can confuse the results. Future work should investigate how to model for such an occurrence in order to compensate for it. Also, a decent resolution is required to accurately estimate the heel strike and toe off angle. Exploration on how to circumvent this limitation can extend the approaches usability in real world environments. Lastly, another possible study is on how alternative limb detectors can be used for this application. Recent developments in deep neural networks have made numerous advancements in estimating a subject's pose in the wild. An evaluation of how these methods (e.g. [12]) can be employed for clinical gait assessment can be very beneficial. Also, a comparison between these limb detection based methods and the embedded sensors like the Gaitrite should also

be explored.

CHAPTER 3
3D POINT-SET TRACKING METHOD: AN APPLICATION IN INFANT KICKING

3.1 Introduction

Infant development studies have identified a connection between the spontaneous kicking patterns of infants and their mental development [64]. The random flexion and extension of legs reflect exploratory exercises towards discovering their utility [65]. As the months progress, the articulations become more purposeful leading to meaningful interactions with the environment, and eventually producing the actions required to turn over and crawl [66, 67]. This connection holds for the first five months of life on average [68], implying that analysis of an infant's kicking motions early in life may provide evidence of potential developmental delay.

Studies have also explored the kicking pattern differences of infants and pre-term infants with white brain matter damage [69, 70]. White brain matter damage can profoundly impact an infant, leading to long-term consequences for many aspects of their development, and possibly contributing to the development of spastic diplegic cerebral palsy (CP) [71]. The above cited studies established distinct differences in the kicking patterns of the two groups. Thus, early kicking patterns show promise to serve as indicators of white brain matter damage.

Typically, most children with CP or other developmental disorders are not diagnosed until the age of 2 years [72]. Considering their significance, having a method to monitor the kicking patterns of at-risk infants during the first five months may reduce the time to diagnosis. Such a system would provide the means to evaluate infant kicking patterns and give therapists a quantitative indicator for the need to intervene early, which can improve

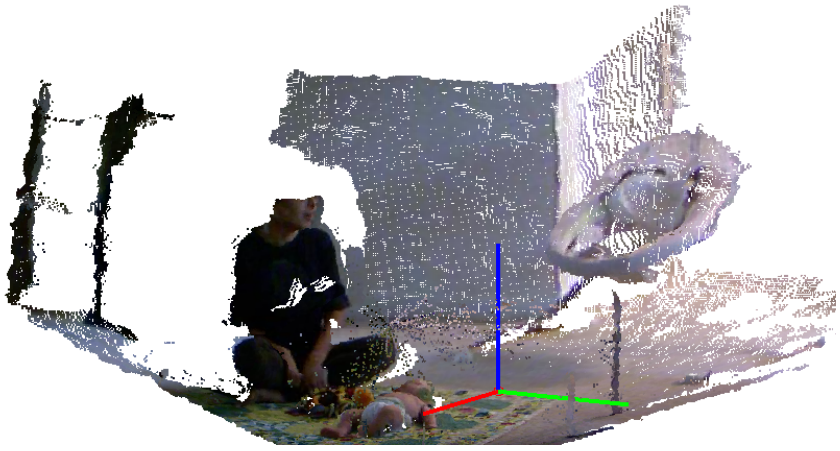

Figure 3.1: Sample colored point cloud, captured indoors from a Kinect, of an infant and their parent at home. Includes a plot of the world frame axes.

the child's quality of life [73].

Typically, studies motion capture systems (mocap) use to monitor an infant movements. Mocap systems work well when markers are strategically placed on the infant's limbs [74]. However, having to place items on the infant may lead to fidgeting and unwanted behavior. This is especially true when the sensors' weight is too much for the infant, impeding their activity and limiting how much of the desired behavior can be observed. This work presents a semi-automated vision based pose estimation system for tracking one leg of an infant. Extending the work presented in [38], the tracker uses an articulated object pose estimator designed for range images. It is augmented by a local geometric measure in the registration formulation, whose role is to improve the system's robustness and increase the basin of attraction of the iterative method. The system works by fitting an approximate, generative model of the subject's leg onto their visual cues using Robust Point Set Registration (RPSR) [37]. Error is accounted for by training a regression model with ground truth observations. The fitting parameters of the generative model provide an estimate of the leg joint angles, and when processed by the regression model, return a final estimate of the infant's leg joint angles. The formulation derived in this chapter will serve as the foundation for the proposed Robust Atriculated Point-set Tracking System (RAPTr).

Tracking the joint angles, an entire video sequence provides an estimate of the baby's kicking pattern. The method is semi-automated, with the user input serving to calibrate the system. It is capable of estimating the joint angles of an infant leg during rapid and chaotic kicking. Furthermore, it meets all of the aforementioned design criteria allowing for at-home sessions. Figure 3.1 depicts the expected setup, where the infant lies with their back on the floor during an indoor capture session.

3.2 Related Works

Tracking the articulated movements of babies is related to the problem of human pose estimation. Human pose estimation seeks to estimate the joint angles associated to a given human body pose [4]. They include approaches with "bottom-up" solutions, identifying the individual limbs to return a final pose optimally fit to them [75] or holistic approaches that process the entire image to estimate the subject's pose. Deep neural network implementations are an example of this [10]. Mostly focused on adults, advances in this area did not always translate to infants. However, recent work has started to investigate solutions for this population [38, 76, 77, 78].

In infant studies, the standard approach is to use a multi-camera, motion capture system configured to track and estimate the 3D positions of infra-red markers placed on the infant's joints [64]. Multiple cameras must view each marker for accurate position triangulation. The requirement that each marker be within the field of view of- and visible by-multiple cameras makes the technique sensitive to self occlusion. Self occlusions occur when the infant's body blocks the marker from being visible by a camera. They limit the range of motion of the legs that can be accurately estimated. Maximizing the range of motion involves installing sufficient cameras at locations determined by the expected marker movements and the associated self-occlusions. Once configured, data collection is reliable and accurate. However, protocols using this class of systems are expensive and involved, due to the cost of the technology, the need for a completely controlled and calibrated space,

and the efforts needed to identify optimal camera viewpoints and marker attachment points on the articulated body. Though considered a gold standard, motion capture systems are unsuited to the design objectives of this work.

Vision based systems relying on a single, monocular camera have been proposed as well. Computing marker positions or pose from a monocular camera is ill-posed without specialized subject-specific knowledge. Therefore solutions focus on measuring movement of the legs. Optical flow, which computes the apparent motion of objects imaged by the camera, has been used to track the relative motions of an infant's extremities. Detection of neonatal seizures [79] and CP [80, 81] is possible through analysis of the computed image velocities. These methods employ visually derived feature vectors as proxies for traditionally used motion kinematics. For applications requiring joint signals or intra limb signal correlations, such as required when assessing an infant kick [64], solely tracking the motion of extremities through a monocular camera will not suffice.

Consumer range cameras, like the Kinect, provide an affordable alternative to complex motion capture systems and create the opportunity for at-home evaluations. From the depth image, simple processing leads to the 3D positions of sensed points in the camera field of view, meaning that only a single such camera is needed. Proper placement of the camera is important to avoid self-occlusion issues, however it is simpler to achieve than in multiple camera setups. Depth based systems have been designed to track the hands and feet of infants by dressing them in specialized clothing [82] or by tracking local extrema with respect to the ground plane [83]. These methods measure the movement of the infant's extremities, as opposed to the individual joints. Estimating the articulated pose of a body requires additional processing. Within the computer vision literature, there are two main approaches for doing so with range cameras [5]. While designed for adult pose estimation, example methods of each type have been proposed recently for infants.

The first approach employs detection-based methods to localize key body parts within the range image. One example utilizes specialized detection algorithms to detect the sub-

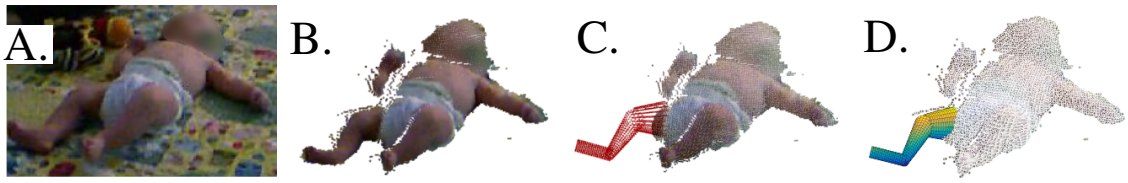

Figure 3.2: A) Input image. B) Segmented infant's point cloud. C) Point cloud mixture model replacing infant's leg. D) Mesh model overlay.

ject's limbs [77]. Another work converts the infant point cloud into a mesh and utilizes the corresponding connectivity graph to measure the geodesic distance between each point to the torso's center [76]. Under the assumption that the hands, feet and head are the furthest away, in a geodesic sense, from the torso's center, the authors then detect the extremities by thresholding. With the centroids of each detected part in hand, both methods use a pre-specified connectivity graph modeling the body's physiology to infer the subject's pose. Graph based methods are limited since their estimates are based on the subject's surface without accounting for the joint offsets.

Model based fitting is the second approach. Presuming the existence of a digital, articulated model of the subject, model-based fitting seeks the joint values that align the virtual model with the visual observations made of the subject's body. Olsen et al [84] initialize the infant's pose by fitting a graph on the surface model's geodesic distance and solving for the inverse kinematics. They then refine their estimate of the final pose by using an iterative closest point (ICP) algorithm. ICP is a hard [85] assignment point-cloud fitting strategy, which is sensitive to missing and extraneous points. An alternative strategy is to use soft assignment algorithms, such as RPSR [38, 86] which does not require an assignment step. We follow this latter approach.

3.3 Methodology

This section describes the proposed semi-automated algorithm for estimating the articulation of an infant's leg over time. Given an input RGB-D image (Figure 3.2.A) and an initial

calibration, the infant point cloud is segmented from the scene clutter (Figure 3.2.B). Next an articulated point cloud model (Figure 3.2.C) whose geometry mimics an infant leg is then fit to the point cloud segment associated with the infant's leg. Estimates of the joint angles are then returned by the fitting optimization. Doing so over the capture's duration while treating the estimate pose from the previous frame as the initial pose for the current frame provides an estimate of the joint signals over time. For comparison, the joint angle signal can be visualized by overlaying a mesh model on the infant's point cloud, Figure 3.2.D.

Importantly, the system supports the presence of a parent or guardian within the testing environment (Figure 3.1). A parent's accompaniment and interaction can stimulate the infant and helps promote spontaneous kicking through play. As the objective is to observe the kicking patterns of the subject, only the lower half of their torso needs to remain visible throughout the capture. Therefore, a parent may remain near the subject's upper half to promote kicking by entertaining them, and to extend capture times by comforting them.

3.3.1 Data Acquisition

Recording of the infant occurs in an open space with their parent or guardian located next to them. Each session commences with the infant being placed on a empty playmat with no clutter that can occlude the infant's appearance. The depth camera, here a Microsoft Kinect camera, is mounted on a tripod in front of and facing the subject, such that it is aimed at the subject's legs with a distance greater than or equal to 1.4 meters (a minimal distance for operating the kinect camera). Spontaneous leg movements are recorded during each trial.

A parent's presence and participation is desired. The parent or guardian is asked to sit at the subject's side. They are allowed to play with the infant in an effort to promote additional kicking and to provide comfort during the session. They are further instructed to not cover the infants leg from the camera's view and to ensure the infant does not roll over. Although it is not common in the early months, rolling over can occur in the later months,

making the capture challenging and giving further reason for the parent's involvement.

The participating family decides the capture duration, ending when either the infant becomes tired or fussy. Also, an adult may elect to stop the experiment. During this investigation, the capture protocol for each infant was run during the mid-morning, typically before their afternoon nap.

The experimental procedures involving human subjects described in this paper were approved by the Institutional Review Board. The child's parents signed the Institutional Review Board approved consent form allowing them to engage in the testing sessions.

3.3.2 Calibration and Subject Segmentation

After acquiring the data, and prior to processing the data, a brief user input phase requests information from the user to calibrate the environment model. This phase uses the input to establish the extrinsic parameters of the camera. The user is requested to delineate sample floor regions in the range image.

Applying principle component analysis (PCA) on the extracted floor patches point cloud data provides an estimate of the floor plane. Since the floor surface is expected to be planar, the normal vector to the floor defines the world z-axis. The up direction is disambiguated from the down direction by selecting the normal vector direction that gives positive z-coordinates for objects in the world (as sensed by the range camera). The x- and y-axes get set by the algorithm. Establishing these world axes with respect to the camera frame gives the orientation of the world frame relative to the camera frame. Figure 3.1 depicts the outcome of calibrating the camera's extrinsic parameters and plotting the world frame within the recovered (colored) point cloud.

Defining the global frame aids the segmentation and tracking of the infant's leg. As the global frame is set on the room's floor and its z-axis is normal to the surface, segmenting the subject is accomplished by thresholding the point cloud z-coordinate in the global frame. After thresholding, the infant should be a dense point cloud surrounded by other dis-

connected point clouds. These other point clouds are from objects within the view of the camera (e.g. clutter, parents, and walls). Calibration of the infant track region requires user selection of the infant's belly to define the target center. A proximity-based connectivity graph seeded from the user selection recovers the infant point cloud. Following extraction of the infant point cloud, the same procedure used to define the global frame, defines a baby frame for the isolated subset of the point cloud. The x/y axis alignment is defined from PCA applied to the infant's planar projection. Defining the eigen-vector associated with the largest eigen-value to be the infant y-axis sets aligns it to the sagittal plane.

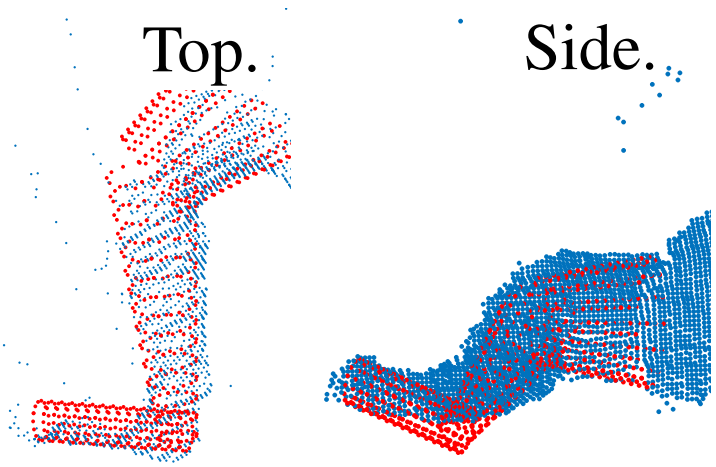

Figure 3.3: Extracted infant leg (blue) and super imposed model (red).

During parts of the capture period, the parent's movements or gestures may place it in close proximity to the baby. In these instances, the parent can be captured as part of the infanct point cloud. Restricting the tracker's view to the lower left projected quadrant (or the lower right) by cropping the data in the other quadrants, limits the point cloud data to that containing the leg of interest, removes the parent's point cloud from the set, and ultimately reduces the algorithm's search space during tracking. This final point cloud is what the model fits to in order to provide an approximation of the infant leg's joint angles (Figure 3.3).

3.3.3 Subject Model and Occlusion Modeling

The infant's leg to be tracked is modeled by a kinematic chain, much like a robotic arm [87]. Each element of the chain is represented using a (conic) cylinder whose form best matches the dimensions of their respective limb. Thus the length and width of each segment must be predefined. A mixtures of Gaussian's model for each limb is created by evenly sampling along the length and circumference of its corresponding cone, with each point denoting the center of a single Gaussian.

The model is defined by its group and shape component. The group component controls the model's global pose and is located at the thigh joint. The shape component is controlled by the revolute joints located at the model's knee and ankle. Adjustment of these joint changes the effective shape of the model, with the ultimate goal being that its shape mimics the subject. Using the formulation presented in [87], the k th leg component is defined by $g^k(\theta) = \prod_{i=1}^{k} g_i(\theta_i)$, where $g_i \in SE(3)$ and θ_i is the component's DoF, and the translation component is defined as displacement equal to the link's length.

Tracking involves generating a hypothesized point-cloud given the infant leg's model and comparing it to the sensed data (as a point cloud). The θ parameters are updated until the infant leg's model matches the subject leg's point cloud.

Occlusions are inherent in range images and must be accounted for. To accomplish consistent tracking with a model-based system, self-occlusion is incorporated in the infant leg model by taking the dot product of the model's surface normals with respect to the simulated camera's optical axis and thresholding for positive values. In other words point i's visibility is determined by $\{p \in X | acos(N_i \dot{N}_C) < \frac{pi}{2}\}$, with N_i the point outward normal to the surface and N_C the camera's optical axis vector. Thus if the surface normal is directed towards the camera (e.g. negative dot product) it must be visible (Figure 3.3.3). However, if there is another closer object with a normal facing the camera, then this other object is deemed to be visible at the point of projection. In this manner, for a given set of articulation joint angles, the infant leg model created for comparison with the sensed point

cloud reflects what would actually be sensed by the camera.

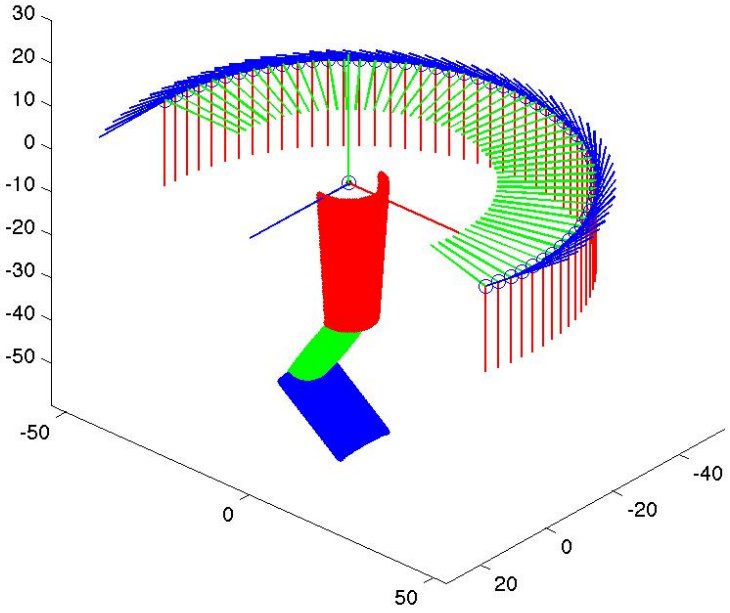

Figure 3.4: Sample image demonstrating how occlusion modeling is accounted for by the system. Only the side of each cylinder visible to the camera is included in the fitting, resulting in the half cylinder shapes demonstrated above.

3.3.4 Robust Point Set Registration

In comparison to ICP, robust point set registration (RPSR) has some advantages. Rather than imposing an association between points (i.e., hard assignment), it uses a radial basis function to identify a zone of influence whose value will be referred to as a soft assignment. Multiple points can be candidate matches, however locally the set of model points that match best with a nearby set of sensed points end up having a greater influence on the estimation outcome. A differentiable radial basis function should be selected for this purpose.

For this application the Gaussian function is selected for it is continuously differentiable and provides near infinite support. This provides the method with a larger region of attraction (versus ICP). These items reduce the complexity of the solution and aid in ensuring convergence even in the presence of large displacements. However, since the solution

is gradient-based and there are local minima in the matching function, the initial estimate should be in the basin of attraction. This is something to be taken into consideration when initializing the model.

Detailing further, the formulation treats both the finite set of points associated with the subject point cloud S and the model point cloud X as mixtures models f and g, respectively. Given a point set $\mathbf{X} = \{\mathbf{x_k}\}$, where $\mathbf{x_k} = \{\mathbf{g_k x_i}\}_{\mathbf{0}}^{\mathbf{N_k}}$ represents the N_k points associated with the kth link, one can define a mixture of Gaussians for the articulated model as

$$f(x, g(\theta)) \;=\; \sum_{k=1}^{N_g} \sum_{i=1}^{N_k} \alpha_i \phi(x, g_k \mu_i, R_k \Sigma_i R_k^T), \tag{3.1}$$

where N_g is the number of links, N_k and R_k represent the number of points and rotation matrix for the kth link. As the subject density is defined by

$$h(x) \;=\; \sum_{j=1}^{N_S} \beta_j \phi(x, \nu_j, \Gamma_j); \tag{3.2}$$

where N_S is the number points in the subject's point cloud and b_j is a scalar term, the pose estimate is accomplished by solving for the following objective function,

$$E(f, h, \theta) = \underset{\theta}{\operatorname{argmin}} \int_{\mathbf{x}} (f_\theta^2 - 2 f_\theta h + h^2)\, d\mathbf{x}. \tag{3.3}$$

The function presented has two invariant terms and one term whose magnitude increases when the fit is good making the aforementioned function equivalent to optimizing the following equation:

$$E(\theta) \;=\; \arg\max_{\theta} \int_{\mathbf{x}} f_\theta h dx \tag{3.4}$$

$$\simeq\; \arg\max_{\theta} \sum_{\mathbf{x}} f_\theta h \tag{3.5}$$

As both the functions f and h are Gaussians, their product is also Gaussian. The resultant function of their product is

$$E(\theta) = \sum_{k=1}^{N_g}\sum_{i=1}^{N_k}\sum_{j=1}^{N_S}\hat{\alpha}\phi(x,\hat{\mu}_{ijk},\hat{\Sigma}_{ijk}) \tag{3.6}$$

$$\hat{\mu}_{ijk} = g_k\mu_i - \nu_j \tag{3.7}$$

$$\hat{\Sigma}_{ijk} = R_k\Sigma_i R_k^T + \Gamma_j \tag{3.8}$$

$$\hat{\alpha} = \alpha_{i,k}\beta_j; \tag{3.9}$$

allowing for the pose estimator update to be defined by the derivative of its explicit representation.

The gradient update is

$$\frac{dE}{d\theta}(\theta) = \sum_{k=1}^{N_g}\sum_{i=1}^{N_k}\sum_{j=1}^{N_S}\hat{\alpha}\phi(x,\hat{\mu}_{ijk},\hat{\Sigma}_{ijk})\hat{\mu}_{ijk}^T\hat{\Sigma}_{ijk}^{-1}\frac{\delta A_k}{\delta\theta}J_k \quad , \tag{3.10}$$

with $\frac{\delta A_k}{\delta\theta}$ denoting the link twist derivative and J_k the Manipulator Jacobian.

3.3.5 Defining $\hat{\alpha}$

Including an influence function within the fitting strategy can help guide the links towards the regions they are designed for. In this work the geodesic distances are used for this purpose. Each point on the subject's point cloud adopts its respective geodesic distance from the infant's center. In case of the model, a reference point is located at the thigh's base. Setting the seed point in this manner ensures a monotonically increasing value along the leg, with the toe gaining the maximal value. To ensure an adequate mapping is achieved, the geodesic values are then normalized by the max value so that the maximal distance is set to one. A sample representation of the geodesic values both the model and infant point cloud are presented in Figure 3.5.A.

Estimating the geodesic distances on the point cloud's surface requires the construction

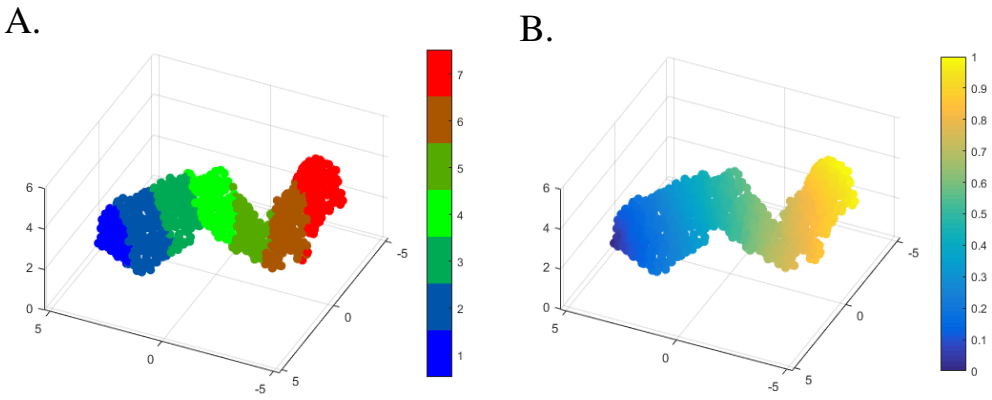

Figure 3.5: Geodesic Distance representation along the leg's length, with the largest values located at the toe and the thigh keeping the lowest (A), Classes labels corresponding to their respective domains(B)

of a connectivity graph [88]. By considering every point as a node, a graph is created by first establishing an inter point proximity measure as follows:

$$A_{i,j} = \begin{cases} d(x_i, x_j), & \text{if } i \neq j \ \& \ d(x_i, x_j) < \mathrm{d}_{max} \\ \inf, & \text{otherwise} \end{cases} \tag{3.11}$$

with the function d defined by the user. For this application, the L_2 norm of the difference is used. All points considered too far from any other point from the graph are ignored.

The resultant matrix then represents the cost of traversing from one point to another. A connectivity graph is defined as:

$$con_{i,j} = \begin{cases} 1, & \text{if } A_{i,j} \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

with τ defining the connection threshold. In order to prevent unrealistic crossover, a conservative value for τ should be defined. This will restrict points on the front from getting connected to another point on the back. A major assumption here is that the captured point cloud is a sparse approximation of the leg's surface, thus using a conservative value for τ

returns a series of superficial connections.

Given the approximate surface, geodesic values at every node are estimated. Using Dijkstra's algorithm at each node with the goal set to what will be considered as the closest point to the object's center (base of the thigh for the model and the belly's center for the infant). Dijkstra's algorithm provides the optimal path from any point to any other point given the cost matrix defined in Equation 3.11. However, it also provides the cost for traversing said optimal path r. This value creates a mutual relationship between the model and subject .

Utilizing an intermittent representation $\hat{r}$ of the calculated values provides a desired mapping for the purpose of model fitting. There is a clear relationship between the model and the subject as the r values are normalized ranging from zero at the base of the leg to one at the toes for both the model and subject. However, r cannot be used directly due to noise present in the range data. Noise creates misleading patterns in the local geodesic values that can confuse the model fitting. To mitigate this problem, the r values are replaced by $\hat{r}$, a class label designating the domain in which the point falls in. Dividing the range from 0 to 1 into equal parts, creates sub sections. Labeling the first section 0 and incrementing the label along the rest provides class values for the points as seen in Figure 3.5.B. It should be noted that the more sections the domain is cut into provides a finer section definition but also allows for noise to have a stronger impact. However, a coarser representation would reduce the established mapping's benefit. Thus there is a balanced number of cuts that will provide the desired fitting. Seven cuts were used in this application.

By defining $\hat{\alpha}$ with local similarity score, the fitting formulation's behavior changes in a manner that directs model regions to their corresponding subject locations. This is the case as the same policy for discretizing the sub sections of the model is employed on the subject as well, ensuring a mutual relationship. The class labels in $\hat{r}$ are not independent as the values indicate an order from base to toe. Thus a difference can be defined to establish a proximity of each label to another converting Equation 3.10 to :

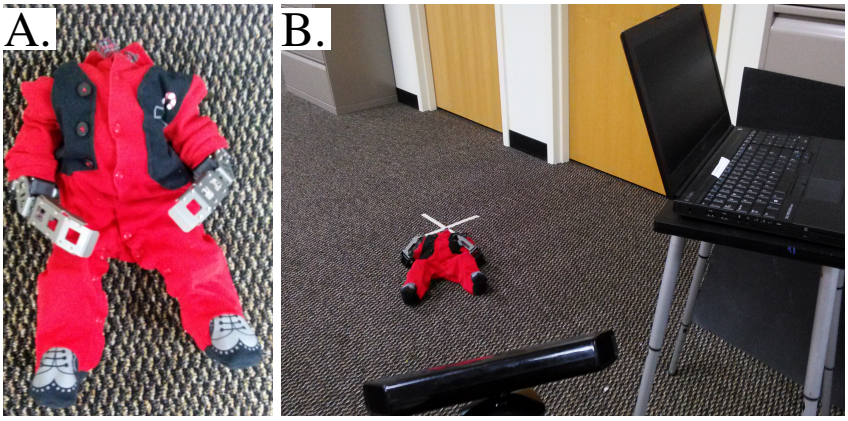

Figure 3.6: "Robo-Baby" (A) and a sample image of the capture protocol (B)

$$\frac{dE}{d\theta}(\theta) = \sum_{k=1}^{N_g}\sum_{i=1}^{N_k}\sum_{j=1}^{N_S}\hat{\alpha}\phi(x,\hat{\mu}_{ijk},\hat{\Sigma}_{ijk})\hat{\mu}_{ijk}^{T}\hat{\Sigma}_{ijk}^{-1}\frac{\delta A_k}{\delta\theta}J_k \tag{3.13}$$

$$\hat{\alpha} = e^{-\sqrt{\frac{(r_{i,k}-r_j)^2}{\lambda}}} \tag{3.14}$$

with λ a scalar defining the bandwidth for r. Including this term in the formulation effectively adjusts the gradient weights, modifying their contribution to the update and ensures like points in their r value map to each other.

There are a couple of benefits to this formulation. Firstly as the gradient magnitudes now adapt based on the proximity of like points in r, the effective region of attraction is increased. Next, once the model is adjusted in a manner that like points in r are close to each other, the RPSR portion of the formulation allows for the local geometry to refine its pose. Lastly the tracker's robustness has increased due to two factors. The first given is that the geodesic distance is semi-invariant to changes in the subject's shape and the second is that using the Mahalanobis distance in the similarity formulation allows for some error in the partitioning.

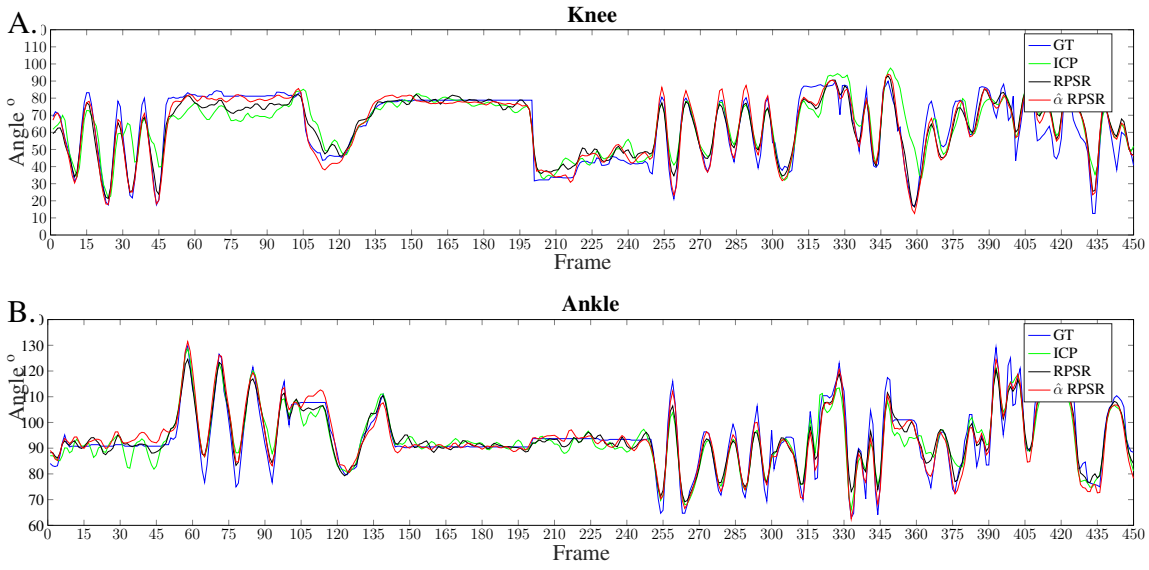

Figure 3.7: An evaluation of the tracking results is presented for the knee (A) and ankle (B). ICP in green and RPSR in red, plotted over the ground truth signal in blue.

3.4 Results and Discussion

3.4.1 "Robo-Baby"

A robotic humanoid is used to evaluate the method's performance. It was constructed in order to mimic the physiology of an infant (Figure 3.6.A), with the capability to enact various kicking motions that mixed both in phase and out of phase flexion and extension patterns in the leg joints. It was designed to have two motors at the hip and one at the knee and ankle joint. Programming of the kicks involved the user actuating various kicking patterns and having the motor record the signal. Various trials of this action were recorded and the resultant kicking signal used for the analysis is a mixture of these recorded trials is a random concatenation of these sessions, tying the tail end of one session to the beginning of another. The final kicking signal (solid lines seen in Figure 3.7) was then actuated by the motors, and recorded using the data collection protocol presented in section 3.3.1. A sample image of the capture protocol is presented in Figure 3.6.B.

3.4.2 Quantitative Error Analysis

Evaluation of the systems performance was done by tracking the leg joints of robot whose shape and dof mimicked the physiology of an infant. A benefit of constructing the "Robo-baby" is that it provides direct access to the actual joint signal enacted during the given frame, facilitating the error analysis.

A calibration protocol must be undergone in order to ensure a valid comparison. The calibration step applies a linear transformation that maps angles to the signal input provided to the robot. This is done to simplify the input commands for articulating the robot, making it so the user only needs to pass in angles. Moreover, using the that logic, the fitting model is designed to work with the same input angles. Thus the model will mimic the robot for any given set of angles and allows for the input signal to be defined as the ground truth.

With the calibration ensuring a direct comparison between the model and robot, estimates of the joint signals measured by the system are compared to the original input joint signals undergone. The resultant error analysis is presented in Figures 3.7.A-B and 3.8.A-B. The error is evaluated using the L_2 norm of each joint signal for every instance in time corresponding to the ground truth signal.

Various kicking patterns were incorporated in the experiments and captured using the protocol presented in section 3.3.1. Nine sessions of kicking actions were collected, including a variety of flexion and extensions of the legs at each joint. The sessions lasted a duration of 30 seconds during which only 50 frames were evenly captured throughout the session. Different types of kicks ranging in magnitude and type were enacted. Effort was taken to prioritize each joint during the first sessions, with remaining session serving as examples where articulation occurred in each joint. For example, the first session prioritized the knee articulation while in the second priority was given to the ankle. Limits in the articulation were enforced, with the robot's effective range of motion for the knee being defined as $[0°, 135°]$ and $[-45°, 45°]$ for the ankle.

The resultant system performance demonstrates its capability to both capture kicking

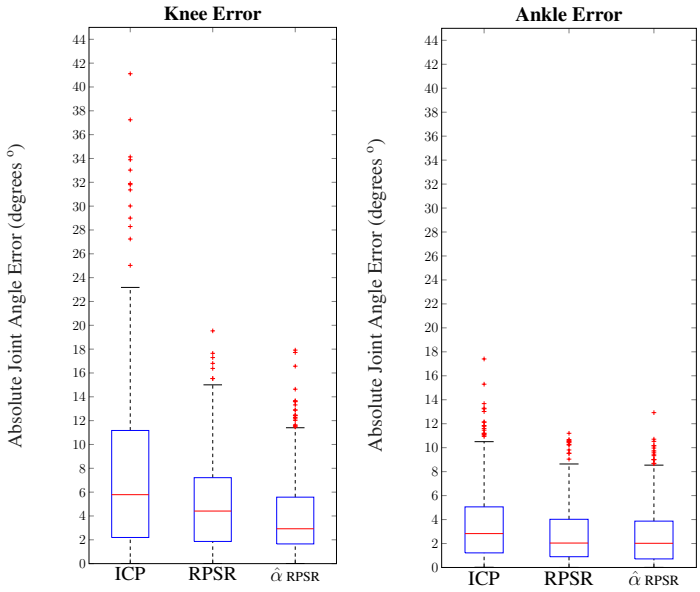

Figure 3.8: Boxplots comparing the results of the ICP and RPSR methods. In both joints, RPSR has a lower tracking error than ICP with a smaller variance as well.

motion trends and return relatively accurate joint angle measurements. In the prior study, presented in [38], the average error was 2.5 and 2 degree error margin for the knee and ankle, respectively. With the new sets' inclusion, this is error grew to 5.1 and 2.62 degree error on average, respectively.

By enhancing the RPSR formulation with the introduction of $\hat{\alpha}$, the system is able to track throughout all the collected sessions at a lower error rate. A major source for the added robustness was due to the $\hat{\alpha}$ associations as they gave priority to the view points that remained visible at the thigh and tibia, ensuring that the corresponding limbs of the model were directed towards them.

A comparison between $\hat{\alpha}$ RPSR, RPSR and ICP (method employed in [84]) is presented. All methods are able to track the robot's motion throughout the nine sessions as seen in Figure 3.7 and even demonstrate the ability to recover if they deviate too much. However, even in the presence of occlusion, both of the RPSR systems are able to track the robot's articulation without detrimental deviation. ICP (green) performed with an average error of 7.16 and 3.37 degrees for the knee and ankle, respectively, while $\hat{\alpha}$ RPSR (black)

had an error of 4.57 degrees for the knee and 2.58 degrees for the ankle. As previously mentioned, the standard RPSR out performed ICP but had a higher error than the proposed $\hat{\alpha}$ RPSR. In addition to having lower average error, both RPSR methods performed with a lower error variance as demonstrated in Figure 3.8.A-B, implying a reduced chance of returning erroneous estimates. No filtering of the signals was applied to any of the experiments.

Further analysis of the resultant figure demonstrates the system's strengths and weakness. With a 7^o per sec maximal rate of change at the hip, knee and ankle, the system was capable of tracking the articulation with good accuracy. This included mixture of flexions and extensions at each joint. An obvious limitation of the method that still persists even with the inclusion $\hat{\alpha}$ in the formulation demonstrates itself at periods when one joint is held constant while the other is actuated. This can be due to two reasons. Firstly the noise present in the Kinect capture and secondly the formulation adjusting its joints in a manner that minimizes the error of the rapidly articulating link while allowing for the other to take on error.

3.4.3 Qualitative Run

For the sake of demonstrating the system's real world applicability, a session was held during which an infant's kicking patterns were recorded and their leg was tracked. Following the protocol outlined in section 3.3.1, various instances of the infant's kicks were recorded. The capture took place in the living room of the subject's home. A parent was presently sitting next to the subject's side promoting their kicking actions through play. One of the infant's toys was used to provide both physical and auditory stimulation. The kinect was set on a fixed location away from the subject. Figure 3.1 is the 3D representation of the capture.

After the session ended, the system was applied to the captured video, resulting in adequate tracking. The estimated joint signals over time are plotted at the base of Figure

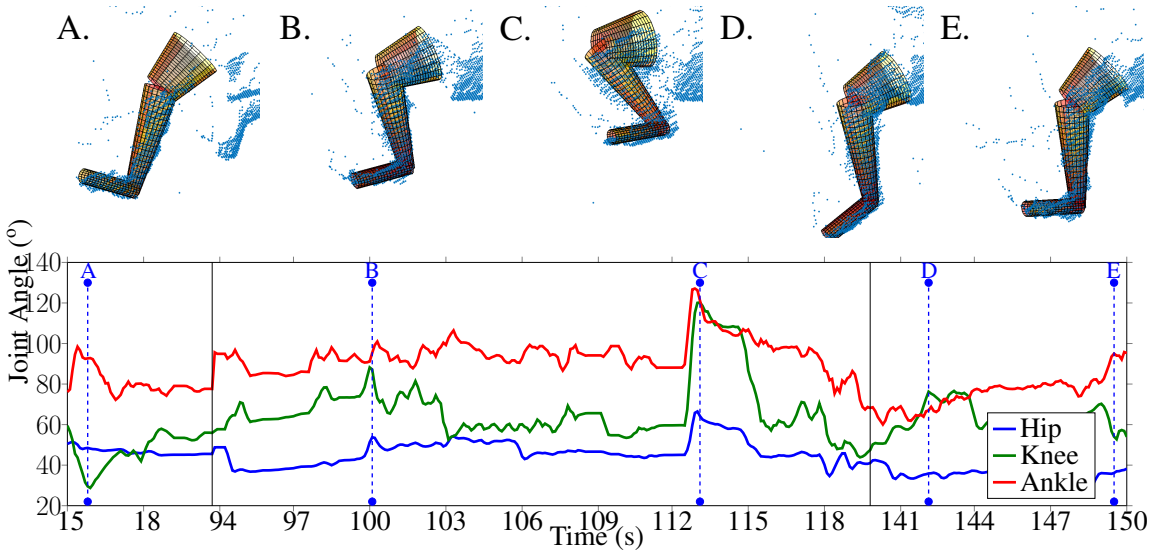

Figure 3.9: Example capture over time. For the sake of this example, the articulated model limb's point clouds is presented by the equivalent mesh. The joint poses captured are presented at 5 points in time. Their corresponding joint angles are marked with blue vertical lines with the blue letter denoting the associated image. Three joint angles over time are presented in red (ankle), green (knee), and blue (projected hip.).

3.9 . Mesh cylinders are used solely for the sake of displaying the model's current pose at the given frame with the blue points representing the subject's point cloud. A letter is assigned to each leg still, marking the frames these poses represent and further noted by the vertical, dashed lines in the graph. As the hip is comprised of a ball and socket joint (having 3 dof), the projected angle with respect to an axis along the sagital plane is plotted in place of the series of joint signal. However, as is evident by the model's superposition over the infant's point cloud, the hip joint estimates are being tracked consistently.

3.5 Conclusion

A semi-automated pose estimation method for an infant's leg was presented. Given some minor manual annotation, the system is able to robustly track the joint trajectories of an infant's leg during spontaneous kicking. It is an extension of prior work in which class labels are embedded onto both the model's and subject's surface, creating an association that

increased the model's region of attraction and improved its capability to track the infant's leg's pose. Its performance is compared to ICP, a standard practice for pose estimaiton. The joint trajectory estimated by this system can be used by a therapist or pediatrician to evaluate the child's development. Furthermore, as the method only requires kinect and computer, it is well suited for at-home sessions.

This work serves as the foundation for the studies presented in Chapters 4 and 5. As the RPSR derivation is so general, it allows for an extension to fully articulated point sets. As will be presented in the following chapters, the formulation allows itself to be utilized for more complex articulated structures, so much so, that articulated model compositions that resemble a human's physiology can be used as well. Treating this as a foundational work, the following chapters extend it for the purpose of doing full body pose estimation of both adults and infants. This is made in large part to the combination of the findings from this chapter and Chapter 2.

CHAPTER 4
ROBUST ARTICULATED POINT-SET TRACKING (RAPTR): FULL INFANT POSE ESTIMATION

4.1 Introduction

Monitoring the kicking patterns of infants gives a very informative glimpse into their development. Tests providing metrics for the infant's health based on their kicking patterns, their arm movements and their neck control inform the identification of early symptoms associated with neurological or physical inhibitors to their growth. Unfortunately, most children with cerebral palsy (CP) are not diagnosed until the age of 2 years [72] due to the subtle indicators that often go unnoticed. It isn't until milestones like crawling and walking are not achieved that awareness of such problems occur. Detection of developmental delays or irregularities in the first months of the infant's life allows for early diagnosis to occur. After which, a regiment of physical therapy or other intervention can be coordinated that when done early can have an impact on the infant's quality of life [73].

In chapter 3, a single leg tracking method was discussed. The formulation presented provides a derivation for the inclusion of variable coefficients. To provide a mapping between the subject and model, a geodesic surface is estimated originating from the thigh's base and extending to the subject's toe. These factors define the coefficient values assigned to each model point and serve to guide the fitting gradients, allowing for smoother and more consistent convergence. Effectively, they create a "soft assignment" between model and subject point, allowing for multiple subject points to inform the gradient at a single model point. Unfortunately, it suffers from one key hindrance. The method is unable to estimate poses where the subject's limbs cross one another well. As presented in [84], the authors identified that this limitation was a consistent source of error. Filtering meth-

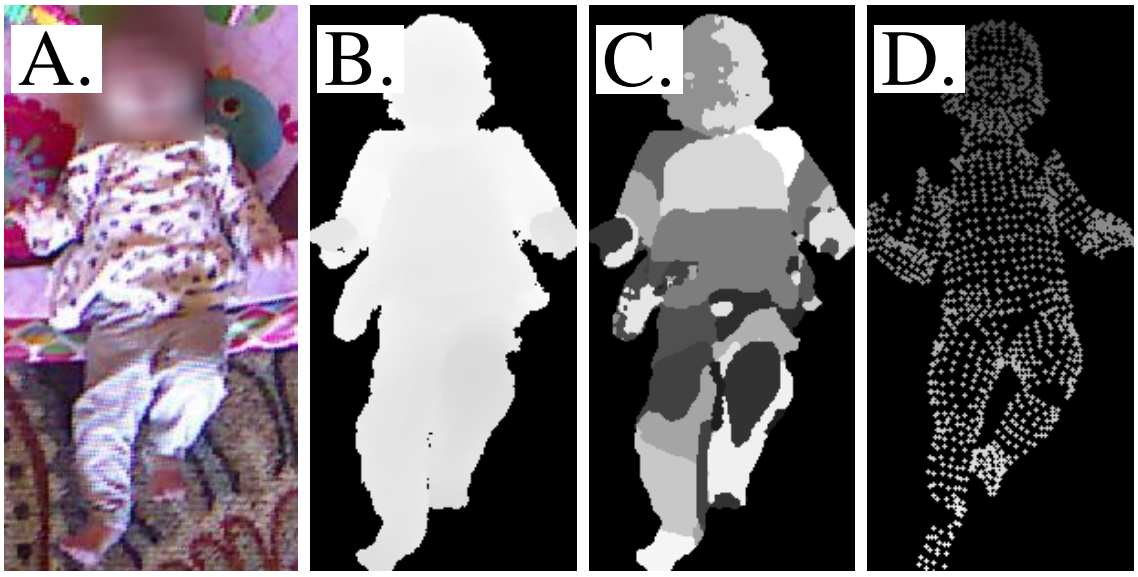

Figure 4.1: A) RGB capture B) range image capture C) classified output, D) pose estimate

ods can be applied to the estimated joint signals alleviate improper pose estimate, but that would violate an intended feature designed for in the proposed solution. It is desired that the approach solve for the pose at every frame, independent of the other frames that came before or after. This is done to reduce the error propagation that occurs in these temporally dependent filter-based solutions [89]. Furthermore, filtering spreads the error across all the fits, propagating error to well-fitted frames.

By giving the limb detectors a series of sample images and their corresponding annotations, any depth image of the subject that resembles the appearances captured in the training data can be processed to return a predicted set of limb location likelihoods. Annotation in this case is the production of images with the limbs colored based on their label. Hence, if a million sample images of different infants can be captured by range cameras and their limbs annotated as prescribed, then such a detector can be trained and the predictions used as a proxy to the geodesic values for defining the coefficients. Although the quantity of images for real infants would be a high barrier of entry to establish this as a viable solution, model-based methods provide an alternative means to accomplishing this task.

The proposed Robust Articulated Point Set Tracking (RAPTr) framework is a bottom-

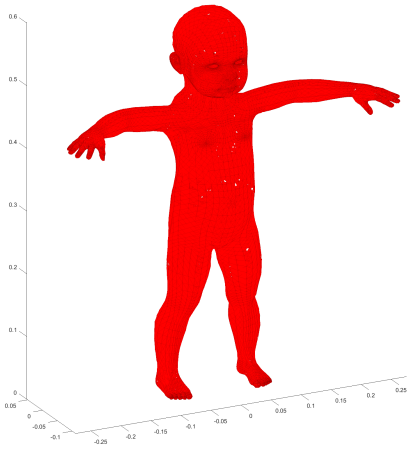

Figure 4.2: Makehuman infant mesh model

up solution that uses an articulated model at multiple steps in the solution to achieve consistent pose estimation. First, the infant model (Figure 4.2) provided by the MakeHuman system [29] is used to create a large number of synthetic infant images and their corresponding annotations. A deconvolution neural network is trained as a limb detector with this set. Next, exploiting the soft-assignments naturally created between the model and subject, an objective function for pose estimation is optimized. Once the pose is estimated, the model's shape parameters are used as shape descriptors to train a final regression algorithm that produces a refined estimate of the subject's pose. Tracking the subject's pose is accomplished by applying this method to every frame of a video capture. Lower error estimates are achieved with this approach when compared to the common practice presented in [77, 78].

This chapter is organized as follows. A related works section will cover some works in infant pose estimation, establishing the solution's baseline. Next, the methodology outlines the model definition, the training set generation, the protocol for defining the coefficients of RAPTr, the model fitting and the shape descriptor-based regression function for the final pose estimate. In the results section, a test for evaluating the algorithm is presented where

Figure 4.3: A) RGB capture B) range image capture: Demonstrating how the subject should appear during the capture

a robotic infant is used to create a ground truth set. Furthermore, qualitative results of a model fit to a real infant are also included. Lastly, the conclusion summarizes the work and possible impact if used towards infant diagnosis.

4.2 Related Works

There have been advances in infant pose estimation through computer vision methods published in recent years. The methods include both loosely limbed and articulated model-based approaches. Each established a baseline for their error metrics. However, no work provided public benchmarking data, public source code, nor binary implementations for reproduction and adequate comparison.

Example articulated model-based methods are available. In [76, 84], the geodesic surface is estimated along the infant's body, originating at the subject's center, leading to the feet, hands or head ending up with the large geodesic values, making the pose estimation problem and inverse kinematic problem. The authors assume that the torso's location is known and employ standard inverse kinematic techniques to retrieve the joint angle initial conditions. They finalize their estimate by applying an Iterative Closest Point (ICP) [90] based update until convergence. The previous work presented in chapter 3 is an extension

of [38]. Instead of an ICP, the method applies RPSR. Unfortunately, both methods suffer from the same limitation: they are unable to cope with poses that arise when the subject's limbs are crossed. This shortcoming is also presented in their findings.

Bottom-up approaches have been applied to infant pose estimation as well. In [77], the authors use the Makehuman system to generate a collection of infant renders. They employ a range imaging model to mimic how the infant is viewed when captured by a Kinect camera. These captures are created using a simulated environment based on certain assumptions. One, the camera is set a known fixed distance away from the subject and two, the model's dimensions (e.g. link lengths) match their subject's dimensions. These captures are used to train a randomized decision ferns limb detector. Furthermore, the predictions are clustered and their centroids are treated as the joint locations. Once more, this is a loosely-limbed method, whereby the limb connections are predefined, resulting in a graph-like model. Also, only the average of each prediction set is treated as the joint location, even though they noted that multiple cluster centroids are possible outcomes. This work is extended in [78], with the inclusion of a cost-based approach to optimize the possible connections that make up the pose, thus reducing the confusion presented by multiple viable cluster centroids. In this work, the output modes calculated by this approach are used as moment-based features for training a linear regression model like [7] did for human pose estimation, which will serve as the method to compare against.

Another sample of a bottom-up approach is presented in [91]. Using a stereo camera, the authors tracked markers which are pasted on the subject's limbs. Only the extremities are marked, constraining the scope of the study to getting the limb's gross general movement.

As mentioned before, each method employed has its own means to establish their error without providing a publicly accessible set for comparison. In [84], a few frames from multiple captures are manually annotated. Treating range images as the median for defining ground truth has a natural limitation as the only points that can serve as markers are those

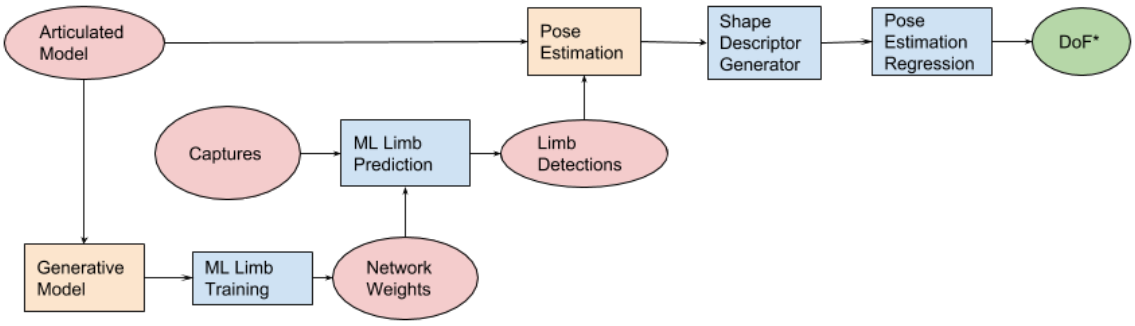

Figure 4.4: High level flow-chart of the proposed RAPTr Framework for Infant Pose Estimation

captured on the subject's surface. This is a problem as occlusions, human error and the fact that the joint lies well below the surface can prevent accurate annotations. In [77, 78], a similar approach is pursued. However, in this instance the authors fit their model to the subject and manually update it till the model appears to correctly match the subject's shape. Once more, this approach is vulnerable to human error.

4.3 Methodology

This section describes the proposed RAPTr framework, an autonomous method for estimating the articulation of an infant on a per frame basis. Given an input depth image Figure 4.1.B, a predicted limb detection map is produced (Figure 4.1.C). Next, this range image is converted into a point-cloud (Figure 4.1.D) and an articulated point set model that mimics an infant is fitted onto it ((Figure 4.1.E), returning an estimate of the infant's pose. Doing so over time, produces an output signal which captures the articulated patterns that make up their actions. The workflow is demonstrated in Figure 4.4. A visual demonstration of the infant subject's actions can then be reproduced by using the estimated pose parameters on the infant model.

4.3.1 Data Acquisition

During a session, an infant's actions are captured from a single range camera. When the session begins, a parent or guardian is asked to set aside an open area on the floor in their home. A mat is set and, ideally, a toy is present. The child is placed on the mat, with the parent or guardian at their side as demonstrated in Figure 4.3.A-B. The range camera used in this exercise is a Microsoft Kinect. It is elevated at least 1.4 meters from the floor using a tripod and pointing orthogonal to the floor. The selected elevation is needed to ensure the subject is within the operating distance of a Kinect camera. This is because if the subject were any closer, the current calibration methods would fail as the range values returned fall into the region not modeled by the calibration functions. Care must be taken to ensure that there are no occlusions present. Although the proposed work is robust to noise, it is still beneficial that the infant remains largely visible throughout the capture duration.

As stated in the previous chapter, a parent's presence is positive and desired. The parent can provide motivation for the child, promoting their activity and keeping them calm during the capture session. Furthermore, their presence affords them the ability to monitor the infant's comfort. Ultimately, it is the parent's determination as to whether the capture should continue or not, making their participation highly important.

The experimental procedures involving human subjects described in this paper were approved by the Institutional Review Board. The child's parents signed the Institutional Review Board approved consent form allowing them to engage in the testing sessions.

4.3.2 Calibration and Reference Frame Definition

Using the Kinect functionality provided by ROS (Robot Operating System), registered images of infant are collected. The ROS libraries [92] can control a Kinect, returning registered RGB and range image pairs. The per-pixel range values are returned in cm, removing the need to calibrate the captured imagery.

Using the intrinsic parameters returned by ROS, the range images are easily converted

to point clouds. First, the x and y index values of each pixel are treated as the projected x,y coordinates of each pixel. Next, each point put in homogeneous notation and multiplied by the inverse of the intrinsic matrix. Lastly, the point are multiplied by their corresponding range value, resulting in a point cloud. The mapping between the pixel and their corresponding points is kept, to allow for predictions in the range image to translate to point cloud.

To remove the features not modeled by the dataset, a reference frame is defined. With the environment being set up for easy capture, the majority of unwanted material in the image can be taken out using standard computer vision techniques. The first to be removed are the depth values associated with the floor. The frame is estimated using a planar model detection method built into Matlab [93]. It identifies the largest set of points that fit a planar model. These points are then treated as observations to solve for a plane to model the floor. Using PCA (Principal Components Analysis), returns a reference frame whose z-axis is orthogonal to the floor. By applying the inverse group product to the point cloud, the resulting points are defined with respect to a reference frame on the floor. Thus, a height threshold can be applied to remove the points associated with the floor. Based on experiments, a 1-2 cm threshold appears to be enough for consistently removing the majority of the floor.

Next, a radius threshold is required to be defined prior to the capture. It is the radius at which any points outside of it are removed from the point cloud. Assuming the child is at the image's center, applying this radius-base threshold removes the point cloud captures of the parent, toys and any other artifact present in the environment that are not associated with the infant.

The reference frame calculated in this step serves the additional function for placing the infant model. With the estimated reference frame, the camera's extrinsic parameters are calculated. Using these parameters, the infant model can be placed directly in the point-cloud capture by setting the model's positional group component terms equal to the reference frame's center.

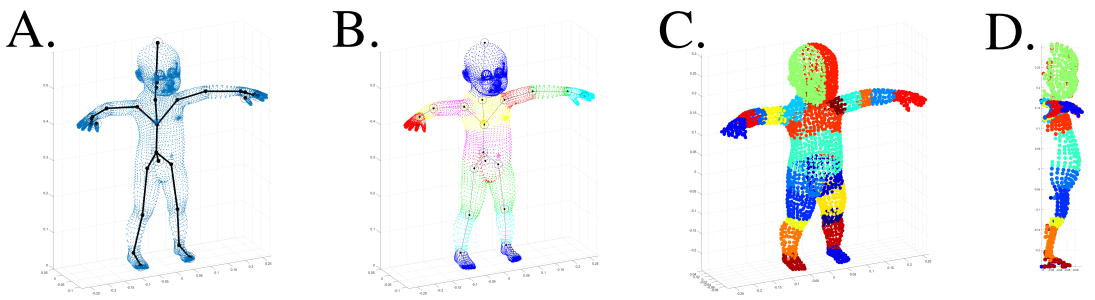

Figure 4.5: A) Model Skeleton, B) Per Limb Mapping, C) Class Defined Mapping, D) Occlusion Modeling [camera in front]

4.3.3 Infant Model

The RAPTr system requires a representative surface model that appears just like the subject. In this instance, the surface model is represented by a point cloud articulated model is used. The Makehuman system has the capability to export a skeletal frame of the infant model (Figure 4.5.A), a mapping of points to their links, the surface normal and a color mapping for each point. A per-link color-coded representation of the infant is presented in Figure (Figure 4.5.B). For the purpose of demonstration, the colors have been randomly assigned to a given link. Each point is been rigidly assigned to their respective links, setting their reference frame at the base joint.

Although the model is structurally different than the one presented in chapter 3, theoretically it still fits the model fitting formulation. The model covered in chapter 3 is a single kinematic chain. In this case, the infant's articulated model is a collection of kinematic chains which are connected to single a base that has 6 DoF group component allowing it move any where in 3d space. However, as each link is its own end effector and is not closed-loop, the formulation from Equation 4.1 still pertains to this fitting objective function.

Several shape Degrees of Freedom (DoF) are provided to the model. Each angular value controls an aspect of the infant model's effective shape. Theoretically, when the

infant model's shape matches that of the subject, an approximation of the pose is achieved. Thus, an appropriate number of DoFs have been included in the model to provide it enough freedom to match the majority of an infant's poses. For this model, 35 DoFs have been defined, with three angular degrees of freedom at each shoulder, hip joint, wrist, ankle, pelvis and neck and lastly one DoF at each elbow and knee. Each joint included in the articulated rigid body model is denoted by a large black marker in Figure 4.5.A.

The six parameters controlling the 3d position and orientation, otherwise known as the group DoFs, are defined to place the infant model over the point cloud capture in what will now be referred to as the capture space. Initial placement of the infant model's position and orientation within the capture space are defined based on the estimated extrinsic parameters and limb detector predictions. Specifics over the shape component initialization and approximate group orientation are covered in Section 4.3.6.

Each point on the model's surface is assigned a label. This represents which class that point belongs to. Figure 4.5.C-D is an example of how the labels are distributed across the infant model's surface. In total, there are 24 labels. These label assignments will later used to generate the annotated imagery.

Occlusion is accounted for in this framework. Figure 4.5.D is an example of the occlusion modeling when the camera is set in front of the subject.

4.3.4 Dataset Generation

Using the MakeHuman program, synthetic range image samples of the infant during a capture session are produced. Each sample range image generated is accompanied by its corresponding per-pixel limb annotations. The synthetic images simulate the infant undergoing various poses while respecting a few assumptions. First, no background is included in the samples. Next, the range camera model is placed at 1.4 meters from where the ground would be (behind the infant). Also, no infeasible poses are included. Lastly, penetration of the infant's limbs to where the floor should be is kept to a minimum.

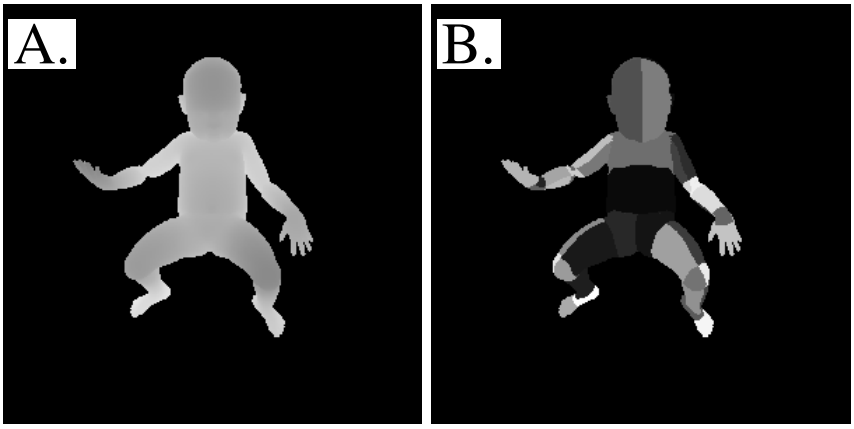

Figure 4.6: A) Synthetic Range Image, B) Per Limb Annotations

Using the built-in range image modeling function and a matte color assignment, a representative sample and its corresponding ground truth image are produced. The range images produced by the Makehuman system adheres to the pinhole projection physics model. Thus each pixel represents a fixed distance, in cm, from the camera with the occluded points omitted from the capture (Figure 4.6.A). A gray scale matte color map is defined, mapping each individual pixel to its corresponding class (Figure 4.6.B). For example, one color denotes the left shoulder while another is set to represent the right. These color assignments serve as the integer class value learned during limb detector training.

A few pose types are defined to ensure a balanced training set is created. Two types of poses are captured. The first type is a set of completely randomly generated poses. The second type is the set of poses created by perturbing from a few initial poses. These poses were selected to serve as initial conditions for creating controlled samples. These poses represent how the infants would look if they were resting, fully extended with their limbs out or if they were in the fetal position. During sample generation, each initial pose is selected randomly from a uniform distribution. A small perturbation is applied to each angle, creating a completely new pose sample while remaining in proximity to the base pose. The combination of the two types of samples produces a balanced set, with both perturbed versions of the common pose set and wildly variable poses evenly present.

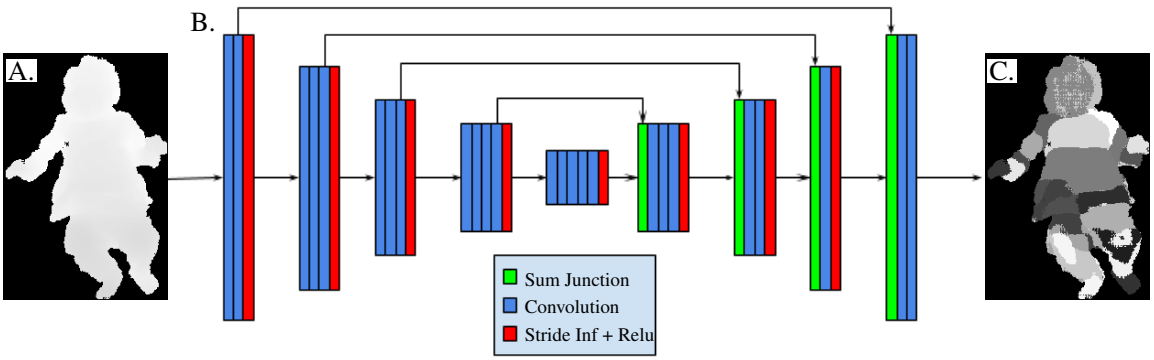

Figure 4.7: A) Range Image Input, B) Deconvolutional Neural Network, C) Classified Output: deconvolutional network with skip connections, stride inference for downsampling and upsampling.

4.3.5 Per-Limb Classification Deconvolutional Neural Networks

Semantic segmentation is accomplished using a deconvolutional neural network. In [94], the authors applied this deep neural network to extract the regions of everyday items like cats, dogs, and or other common objects from imagery. This work was extended for the use of semantically segmenting subsections of humans in [7]. Although other methods of semantic segmentation are available: randomized decision tree [95, 14, 22], fields [96] and ferns [77], this work will focus on the implementation and performance evaluation of deconvolutional neural networks. Comparison between this network's performance and that of a randomized decision ferns implementation are presented in the results section. The randomized decision ferns are selected as the baseline competing method because it is the most recently published limb detection method used for the application of infant pose estimation and also because they demonstrated that the classification results were comparable between the Randomized Decision Forests and Ferns.

An implementation of Segnet [97] is applied to each input range image capture. Segnet is a deconvolutional neural network which returns a per-pixel classification. In this application, it is treated as a limb detector. Using several convolutional operators at each level, the network creates a series of feature outputs that when processed with each layer's Relu

Figure 4.8: A) range image, B) warped range image

operators, returns highly descriptive features. At each level, a downsampling mechanism is applied. Normally max-pooling is selected towards this. However, in this work, stride inference is used. Essentially, it is a means of doing convolutional operations in routine strides by skipping a few pixels along the image at every operation, and thus reducing the dimension of the input image [98]. A similar operation is applied to upsample the output convolutions in the network's latter end.

The first five stacks are structured by the following dimensions: 300x300x64, 150x150x128, 75x75x256, 38x38x512 and 19x19x1024. The first two values are the image dimensions and the last entry is the number of filters. As the network is symmetric, the dimensions of the network's remaining four components are equal to the first four dimension sets presented, just in reverse order.

The Segnet implementation is a variation of the original published work. A series of skip connections, like those implemented in the U-Net [99], are incorporated into the network structure. As demonstrated in Figure 4.7 , they traverse the network connecting the end of earlier convolution stacks to the beginning of their corresponding later convolution stacks. These serve to capture the details from earlier convolution operations that may be lost from the numerous downsampling steps applied to the input image.

Each image passed into the model requires normalization. In this instance, normaliza-

tion refers to the process of mapping the input image into a common space which all input images can be mapped to. The following procedure is applied to accomplish this. First, the images are cropped such that the extents of the crop are defined by the pixel which have non-zero values. Only the range images are considered for defining this extent. Next, once the image is cropped, they are then resized to a canonical image dimension. A height and width equal to 300 is used in this work as seen in Figure 4.8. Lastly, a histogram stretch forcing the range values between a maximum value of 1 and a minimum value of 0 is applied. Although batch normalization is included in the deconvolutional set, this step is still required for the randomized decision ferns. Thus, this has been included in both methods to ensure the same there is a fair evaluation of each method's performance. Also, each step; except for the histogram stretch, is applied to the label images for training. When evaluating the images, an inverse operation of each of the prior steps is required to be applied to the resultant predicted label image, excluding the histogram stretch.

This model serves to provide limb detections. Each image processed by this network, returning a per-pixel classified image as demonstrated by Figure 4.7.A-C. As the model is used to both generate the samples for training and fit to the subject, the mapping between predicted outputs and their corresponding limbs on the articulated infant model are already defined.

4.3.6 Model Initialization

For the fitting formulation to work well, the articulated model must be instantiated close to the subject's pose. Proximity in this case refers to having the model's group orientation and position be near the values of the actual subject's group components. Essentially, the closer the initial conditions are to the actual pose estimate, the shorter the period to convergence. Also, this lowers the possibility of having the optimization function get caught in a local minimum, resulting in a more accurate pose estimate.

Initialization of the articulated model is done by setting the model's group components

based on some heuristic from the capture, calibration parameters and classification results. Using the extrinsic values provided by the ROS package, the infant model is placed in the capture space. The positional group components are set by equating their values equal to the point cloud's center, otherwise known as the point cloud's mean. Next, the group orientation is defined by three vectors. The first vector is the direction established by the infant's point cloud mean to the average of head-class points. The second vector is the estimated floor plane's z-axis. The third and last vector is defined by the cross product of the two estimated vectors. The resultant initial rotation matrix is created by their concatenation which is finalized by applying the Graham-Schmidt process of orthogonalization.

4.3.7 Model Fitting

Treating each point on the articulated model as a Gaussian function and having them rigidly assigned to a link within a fully articulated infant model, allows for the use of the formulation presented in chapter 3. Once more, with f and h denoting the articulated model (Equation 3.1) and subject points (Equation 3.2), respectively. A function modeling the product of two mixtures of Gaussians is defined by

$$E(\theta) = \sum_{k=1}^{N_g} \sum_{i=1}^{N_k} \sum_{j=1}^{N_S} \hat{\alpha} \phi(x, \hat{\mu}_{ijk}, \hat{\Sigma}_{ijk}) \tag{4.1}$$

$$\hat{\mu}_{ijk} = g_k \mu_i - \nu_j \tag{4.2}$$

$$\hat{\Sigma}_{ijk} = R_k \Sigma_i R_k^T + \Gamma_j \tag{4.3}$$

$$\hat{\alpha} = \alpha_{i,k} \beta_j, \tag{4.4}$$

with g_k and R_k being the link's group function and rotation matrix, respectively. As the product of two Gaussians is also a Gaussian and differentiable, the derivative returns an explicit representation for the update.

The gradient update is

$$\frac{dE}{d\theta}(\theta) = \sum_{k=1}^{N_g}\sum_{i=1}^{N_k}\sum_{j=1}^{N_S}\hat{\alpha}\phi(x,\hat{\mu}_{ijk},\hat{\Sigma}_{ijk})\hat{\mu}_{ijk}^{T}\hat{\Sigma}_{ijk}^{-1}\frac{\delta A_k}{\delta\theta}J_k \quad , \tag{4.5}$$

with $\frac{\delta A_k}{\delta\theta}$ denoting the link twist derivative and J_k the Manipulator Jacobian. This derivative defines a direction in the pose parameters space, that can be followed to retrieve an estimate for the subject's pose. The alpha value present in the Equation 4.5 is defined by the label correspondence between the subject and the articulated model for the given point. Although this formulation was originally defined for single kinematic chain, it applies to an articulated object made up of multiple kinematic chains as well.

Using the gradient descent approach, the model's pose is updated based on the gradient direction until it matches the subject's. By applying Equation 4.5 to the model's DoF variables until convergence, an estimate of the pose is achieved. It is by design, that an articulated model is assumed to have converged.

Alpha values are assigned to the articulated model's points in a natural way. The formulation in Equation 4.5 includes an alpha coefficient which can be designed to achieve better convergence. For this application, alpha is substituted by:

$$\alpha_{i,j} = \begin{cases} 1, & \text{if } C_i = C_j \\ 0, & \text{otherwise} \end{cases}, \tag{4.6}$$

with C_i and C_j representing the model and subject point predicted class label, respectively. The resultant gradients end up directing the limbs to their corresponding sections in the subject point cloud. This also results in a larger region of attraction for convergence and introduces some robustness to noisy points, which is a direct result from the formulation being Gaussian, essentially creating soft assignments between a single model point to many subject points, and also producing a robustness to false-positive limb detections.

Originally, the goal was to define $\alpha_{i,j} = p(C_j = C_i|v_j)$, however the use of equation

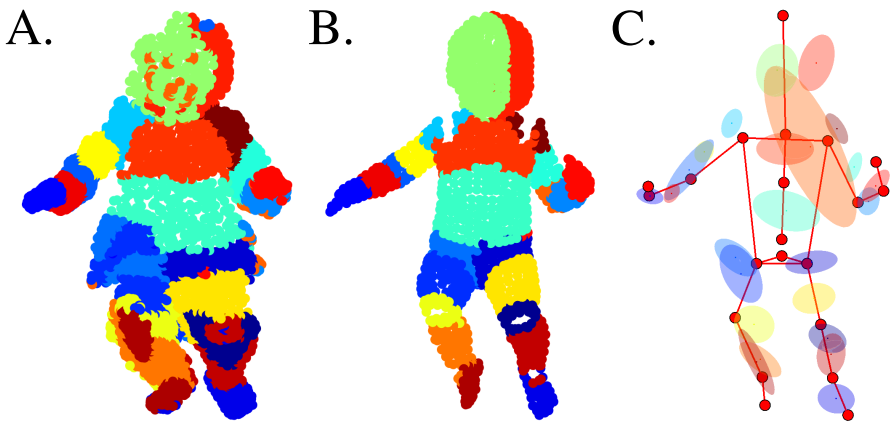

Figure 4.9: A) Limb Semantic Segmentation on a real infant, B) Model Fit, C) Moment-based descriptors demonstrated via colored ellipsoids, where the colors represent the class and the model skeleton is super imposed.

4.6 results in faster convergence and better fits.

4.3.8 Pose Estimation

A regression model is trained to estimate the final pose. In [77] et al, centroids of each group isolated by its predicted label is used to model the closest joint. However, this is not a very accurate approach as the joint locations may reside well below the point cloud surface. Mean values taken from point clouds generated from range images tend to stay around the surface. However, these values can serve as feature descriptor to train a linear regression model.

To that end, both moment-based and shape-based descriptors are calculated in this work. First, a moment-based feature vector is estimated from the subject's point-cloud. For example the subject's classified point cloud depicted in Figure 4.9.A return the moments represented by the ellipsoids in Figure 4.9.C. A vector component is extracted for each prediction cluster, resulting in a vector that is comprised of the mean and standard deviation from class. If no representative points for a given class are available, the descriptor is defined by a vector of zeros matching the intended dimension. This is a common occurrence, as false positives and occlusion are possible. When estimated, these features

provide insight about the subject's shape as they mimic the pose up to a certain extent. The second descriptor is created via the model which serves as a feature generator. Taking the estimated joint parameters and the link end-points as features (From the model in Figure 4.9.B, the end-points are represented in Figure 4.9.C by the red markers), provides further insight into the subject's pose. In the results section, the use of these features are tested to evaluate their utility in training a final linear regression model.

Although alternative regressors are available, a linear regression model was chosen because it is less likely to overfit to the captured data.

4.4 Results

In this section, two methods of validation are presented.

The first method is a quantitative measure of the system's performance. A robotic infant is monitored, and the proposed methodology is applied to estimate the effective pose. The classification performance of both the randomized decision ferns and deconvolutional neural network is evaluated as well. The pose estimates, tested using both limb detector methods, are compared to the ground truth values provided by the Optitrack system. This is done for a series of moment-based and shape-based descriptors, with the shape-based descriptors representing the RAPTr solution.

Next, qualitative evidence is provided. An infant is monitored using a Kinect camera in the comfort of their home. Their movements are captured and processed using the proposed method. Snapshots of the capture with their corresponding pose estimates are presented. For the sake of presentation, each estimate is represented by the fitted infant model.

4.4.1 Classification Error

A comparison between the classification error from randomized decision ferns and deconvolutional neural networks is made in this section. Each model is trained on a collection of randomly generated infant images and evaluated on a subset.

Infant Training and Validation Set

A set of thirty thousand infant images is used for this experiment. The set is generated using the approach presented in section 4.3.4 and consists of 30,000 sample infant images. The synthetic infant model is provided by the MakeHuman API with the limb ratios and height remaining untouched. The camera model employed is set at 1.4 m above the synthetic infant and the output images are 300 x 300 pixels in dimension. As this is a synthetic set, no background is simulated.

A training and validation set are created from the collection of synthetic images. The training and validation set are made up of 75% and 25% of the full set, respectively. Only the training set is used to train the models and only the validation set is used to measure the classification accuracy.

In this experiment, both classifiers are processed on the same images that followed the extraction and normalization protocol presented in section 4.3.5. With samples being synthetic, no background subtraction is necessary. However, normalization, is used on each input image for training and validation. Accuracy is measured on the output images after the normalization is undone, returning the classified image to the original input image's dimensions prior to normalization.

Per Pixel Error Analysis

A direct per-pixel comparison with the ground truth label image defines the classification error. The validation set is made up of a series of sample depth and label image pairs. Each depth image is processed, and the outputs are compared, at the per-pixel level, to its corresponding label image.

The error for each approach is presented in two ways. First, the total error, is presented in Figure 4.10.A. Next, the per-part classification average error is presented in Figure 4.10.B. As demonstrated by Figure 4.10.A, the deconvolutional neural network outperforms the randomized decision ferns by 3.75%. Not including the infant's head, the

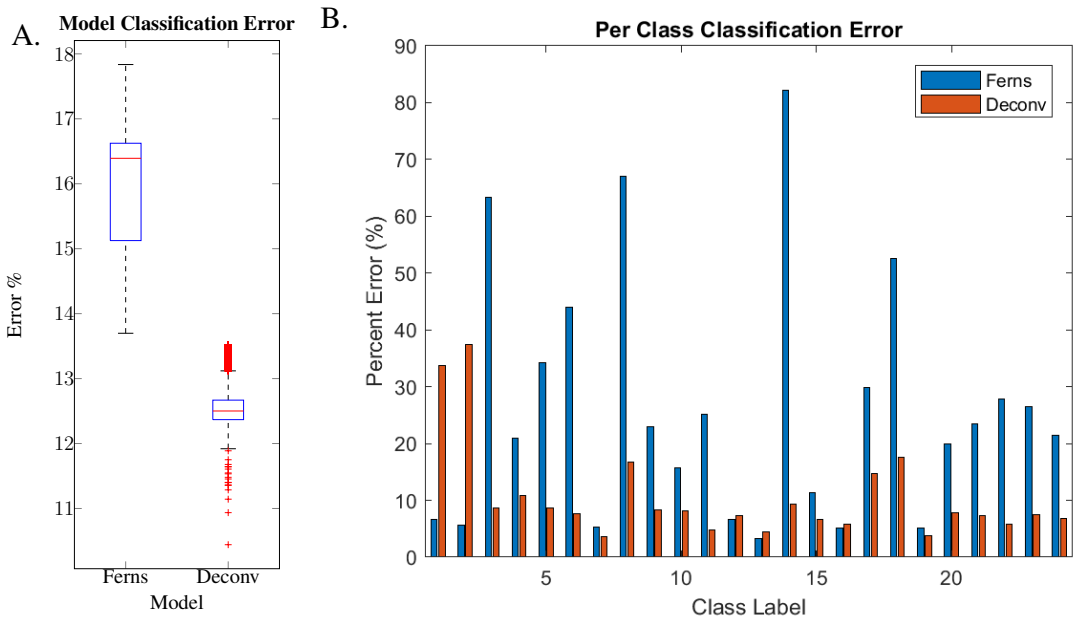

Figure 4.10: A) Average error across entire surface, B) Per class classification error

deconvolutional neural network also manages to get more of the per-limb pixel classifications correctly classified as well. Namely, the deconvolutional neural network is able to detect more knee and elbow labels positively than the randomized decision ferns. Based on the confusion matrix presented in Figure 4.11.A, the deconvolutional network appears to get the head labels confused with one another. However, it also demonstrates that in general the deconvolutional neural network performs better than the ferns model as seen in Figure 4.11.B.

As presented in Figure 4.12, the sample classified images demonstrate the predictions of each method when applied to the same input image. Although only a single sample, the major trend in the difference in the performance of these approaches is captured. First, the predominant error for both methods occurs mainly at the boundaries between class labels. Next, the deconvnet can classify the pixels at the joint better than the randomized decision ferns. This is probably because the deconvnet captures spatial information in the learned filter responses. Ferns, however, return better results around the infant's head. If the average is taken along the average error per part, the discrepancy in their performance

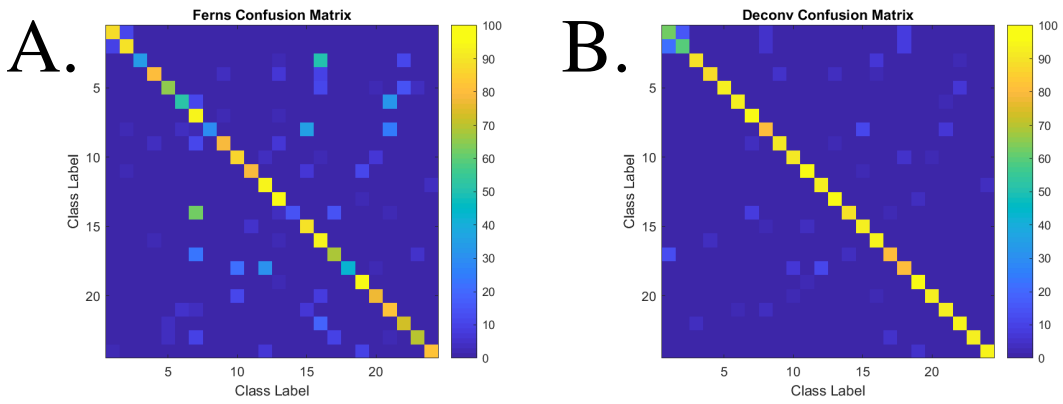

Figure 4.11: A) Confusion matrix for the ferns results, B) Confusion matrix for the deconvolutional neural network results

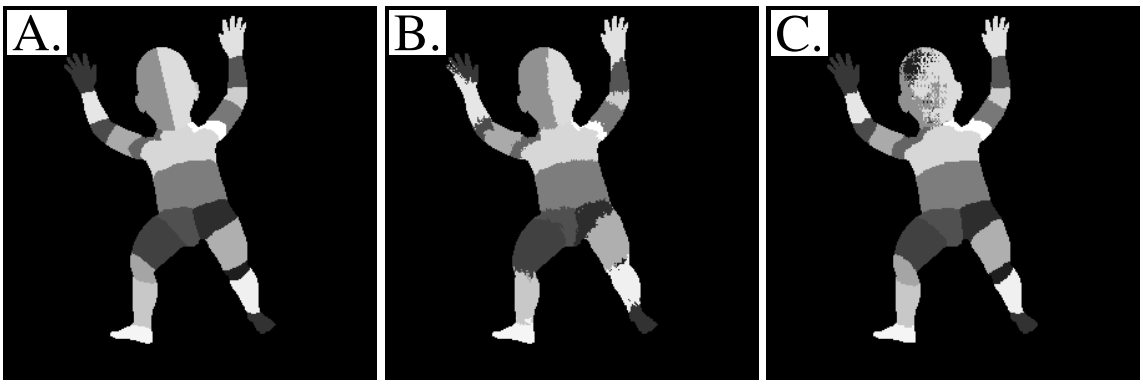

Figure 4.12: A) Ground Truth Sample, B) Ferns Classification, C) Deconvolutional Neural Network Classification

is even more stark. This is the case as the randomized decision ferns fails, on average, to capture the labels at the subject's joints. In [77], the authors used large label regions at each joint. This may have attributed to their better performance at the joints.

4.4.2 Pose Estimation Error

A relatively low L2 error is required for use of pose estimation methods towards medical diagnosis. With the patient's health outcome being so dependent on the clinician's diagnosis, care must be taken to ensure an acceptable margin of error in the pose estimation. L2 error is the measure of choice, as it represents the error of the subject's and their limb's rel-

ative position. From these 3d positional or angular DoF values, a clinician can estimate the required measurements, in their respective space, needed to establish diagnostic evidence. For example, given the subject's joint's 3d position in some frame, a clinician can estimate joint angles. Doing so over time, provides a glimpse in to the subject's movement profile. Depending on the action being evaluated, these signals inform the clinician as to the necessary therapy or intervention required to aid the patient in improving their condition.

L2 error analysis is used in this section to compare the performance of the proposed method with the state-of-the-art method. Additional variants of the proposed method are evaluated, and the results are demonstrated in the following sections.

Robotic Infant

For the sake of validation, a robotic infant is constructed. Mimicking the physiology of a 4-month-old child, the robot has a height of .6 meters, 16 DoFs and a plastic infant head. The robot's torso acts as a base. The head is fixed to the torso and four kinematic chains are connected representing an infant's arms and legs. The DoFs are placed as follows: two joints at each shoulder, thigh and one at each knee, elbow and ankle. Furthermore, to capture the round, deformable and soft appearance common to and infant, the robot is dressed in a one-piece pajama, with socks wrapped around the metallic end-effectors that make up its hands.

At the experiment's beginning, an open space is designated with the robot placed at its center on top of a white mat. The mat provides a simple means to define a background model. With respect to Section 4.3.2, this mat serves as a floor proxy which is easier to extract. Thus, in addition to the planar fit used in the calibration step, a color model is also used to ensure only the robotic infant points are fitted to. Also, an Optitrack system with six cameras is set up. The cameras are placed around the infant at different heights allowing for the inclusion of the range camera. The range camera is situated on a tripod facing the robot from above.

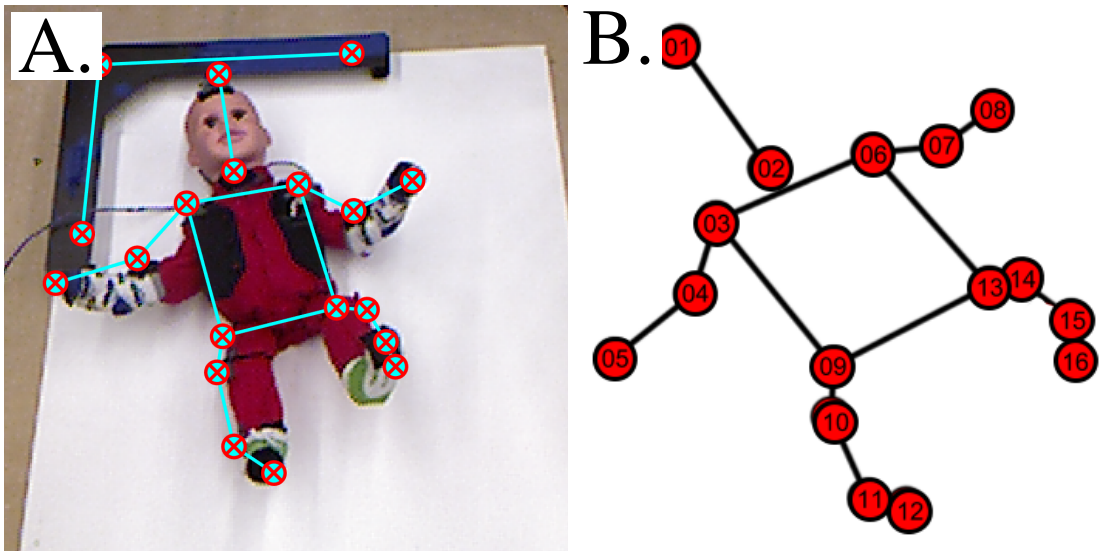

Figure 4.13: Sample infant capture with the ground truth skeleton overlaid

The robotic infant can actuate in a manner like a real infant. With the base not tied to the floor. As the robot actuates it is free to move. Numerous flexions and extensions are applied at each joint, causing several movements that shift the robot slightly per pose. Allowing for the creation of poses where the infant is not perfectly straight. Furthermore, the poses are generated at random.

Using a motor controller driven via ROS packages, the robotic infant is actuated, creating a random set of poses. At even intervals, each of the robot's joints are actuated from a home configuration by a random value. The effective angle of the actuation is limited by predefined hard limits to prevent infeasible poses. Also, every sample pose captured is independent of the last in an effort to ensure temporal independence. This is a typical feature in datasets designed to test a pose estimators performance.

A sample range and monocular image is captured at each interval. With the infant actuated, the Kinect is set to take a single depth image at the point when the robotic articulation converges to its randomly generated pose. The capture returns both a RGB and range image. In combination with the open-source ROS calibration packages, a corresponding

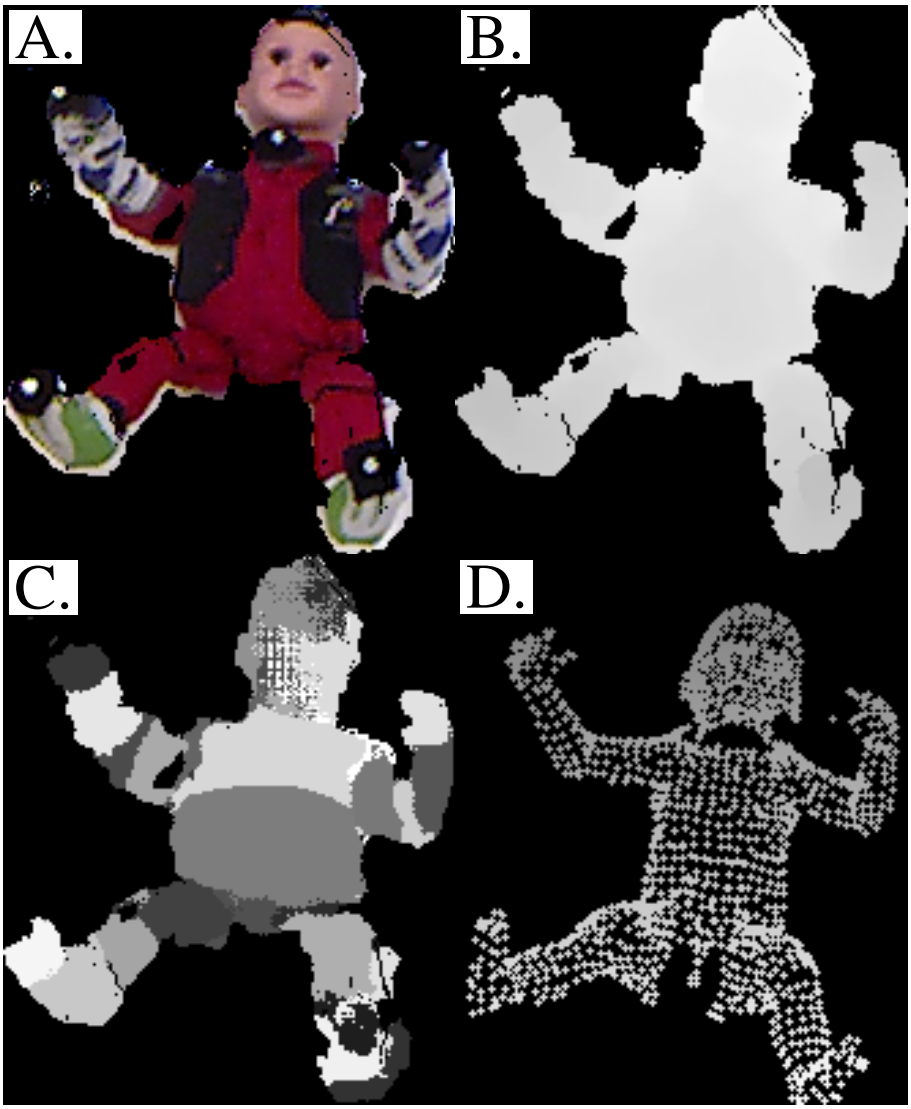

Figure 4.14: A) preprocessed robotic infant, B) range image, C) class prediction, D) model fit

calibrated point-cloud is extracted for each pose.

The ground truth is defined by the Optitrack system. Sixteen markers are placed at the shoulders, elbows, hands, hips, knees, ankles and toes with one placed at the top of the robotic infant's head and another at the neck 4.13. These markers are tracked by the six optitrack cameras. The output estimates are then attributed to their corresponding depth images by matching the capture time-stamps. Definition of the common reference frame is accomplished by solving for the floor plane. Next, the reference frame created by the three points located at the black l-shaped rig is estimated. In Figure 4.13, the rig is located in the upper left area. Lastly, through standard absolute orientation optimization [100] the ground truth points are mapped into the calibrated point-cloud space. Thus, the ground truth is defined by the 3d position of the markers.

An evaluation set is created using the aforementioned steps. A collection of 200 samples with their corresponding ground truth pose estimates are collected to evaluate the RAPTr system's performance.

L2 Pose Error Results

Using the robotic infant presented in Section 4.4.2, a means to evaluate the system's performance based on the L2 error is established and used. The experiment goes as follows. Both the fern-based and deconvolution-based limb detector are applied to each capture. The outputs are then processed to produce a collection of moment-based features as defined in Section 4.3.8. Next, treating the predictions as weights for the model-fitting algorithm, an articulated model is fit to the point-cloud data using RAPTr. Also, both the joint positions and the articulation parameters are collected and treated as shape-based features.

The L2 error results are presented in Figure 4.15. A linear model trained using an 80% training and 20% testing split per run for 100 runs. At each run, the indexing for the training and testing are held constant to ensure every approach tested is evaluated using the same input data. Four sets of features are tested in each test. The baseline method is represented

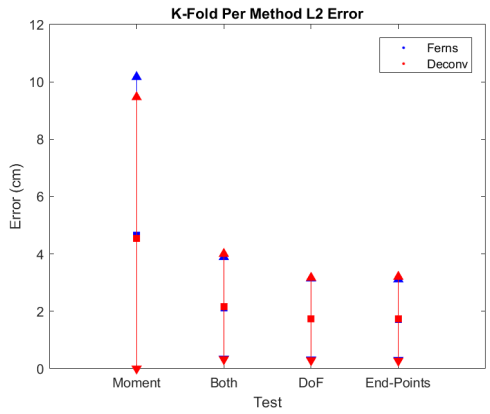

Figure 4.15: L2 Error Statistics

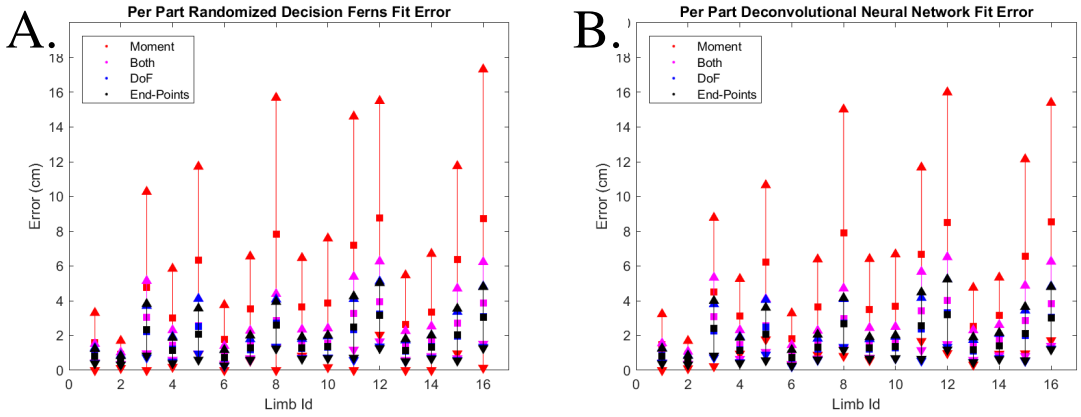

Figure 4.16: Per-part L2 Error Statistics for A) Randomized decision ferns B) Deconvolutional neural networks

by the moment-based features extracted from the output predictions, which are compared to the shape-based descriptor generated using the RAPTr system. These filters are tested as follows: DoF parameters, model end-points and the concatenation of both. Samples of the input RGB and range image pair are presented in Figure 4.14.A-B, the output classification in Figure 4.14.C and the output pose in 4.14.D.

Consistent pose estimation error is apparent regardless of the limb detector used. Evidence suggests that all of the pose estimators work equally well when using different limb detectors, even though there is a clear difference in the performance between deconvolutional neural networks and randomized decision ferns, with deconvolutional neural net-

works having better gross error and per part classification error. This is possibly due to RAPTr's robustness to false positive classification.

From the experiment, evidence suggests that the shape-based descriptors outperform the moment-based descriptors. Errors of 1.74, 1.731 and 2.165 cm are witnessed when using the end-points, angles and their combination as input features, respectively. Each of these values is less than the competing approaches 4.54 cm error. Furthermore, the variance present in each of the shape-based features error is far lower than the moment-based results variance. One possible reason for the shape-based descriptors out performing the moment-based descriptors in these experiments is that because they model the subject's body, the resultant joint estimates end up physically closer to the true joint positions. As opposed to the moment-based features only capture local surface trends. As they are not as to the true underlying joint positions, they are not as informative. Another possible cause for this additional error is that the moment-based features are susceptible to the classification error, while the shape-based features are not as the fitting algorithm has a higher tolerance to classification error.

The last observation demonstrated by the results in Figure 4.16 is that error propagates from the subject's center to their outer most parts (hands and feet). The id definitions are presented in Figure 4.13 Error values measured at the joints closer to the torso are noticeably smaller than error present at the subject's extremities. One possible cause can be that there is far less variation in the inner joint's position as compared to the outer joints. Implying that the range of values needed to be estimated by the regressor is smaller, thus easier to capture than are with high variation. With respect to the model fit, a possible cause of error present at the extremities is that the error propagates from inside out. For example, error at the torso leads to error at the shoulders and hips which leads and so on and so forth until the kinematic chain. This is the case, as each of the segments is dependent on the segment that preceded it. This error is fed directly into linear regressor.

4.4.3 Qualitative Error analysis

Using the capture protocol presented in Section 4.3.1, an infant's motion is tracked using the proposed method. In the presence of the infant's guardian and in the virtue of their home, a sample video capture is collected. The infant is placed on top of a carpeted floor facing upwards, over a blanket. A toy set, which hangs along an arc, is placed near the infant's head. Care must be taken to ensure it doesn't occlude any part of the child. The child is in range of their toys with their parent located beside them. The Kinect camera is mounted on a tripod, over the infant, almost orthogonal to the ground while not being occluded by the child's toy. A video made up of range images is collected, capturing a series of the infant's actions ranging from kicking to are stretches and flexions. A sample frame from the capture is presented in Figure 4.3.

A number of select frames from the capture are presented in Figure

Some pose estimation error was still present at specific times during the subject's capture. In some frames, the subject had crossed their legs. Unfortunately, in a few frames during this period, this led to some confusion on the limb detector outputs. Although, RAPTr is robust to misclassification for internal surfaces, missclassification at the extremities can cause the fitting function to converge to a non-optimal local minimum. The RAPTr algorithm uses the neighboring links to compensate for any confusion present in the internal ones. This is analogous to interpolation. However, when misclassification is present at the extremities, the algorithm uses internal links to compensate for error that the external limbs have incurred. This is analogous to extrapolation, which is known to have worse performance than interpolation. Filtering methods can be employed to improve the results. They were not applied in this instance as it was beyond the scope of the work.

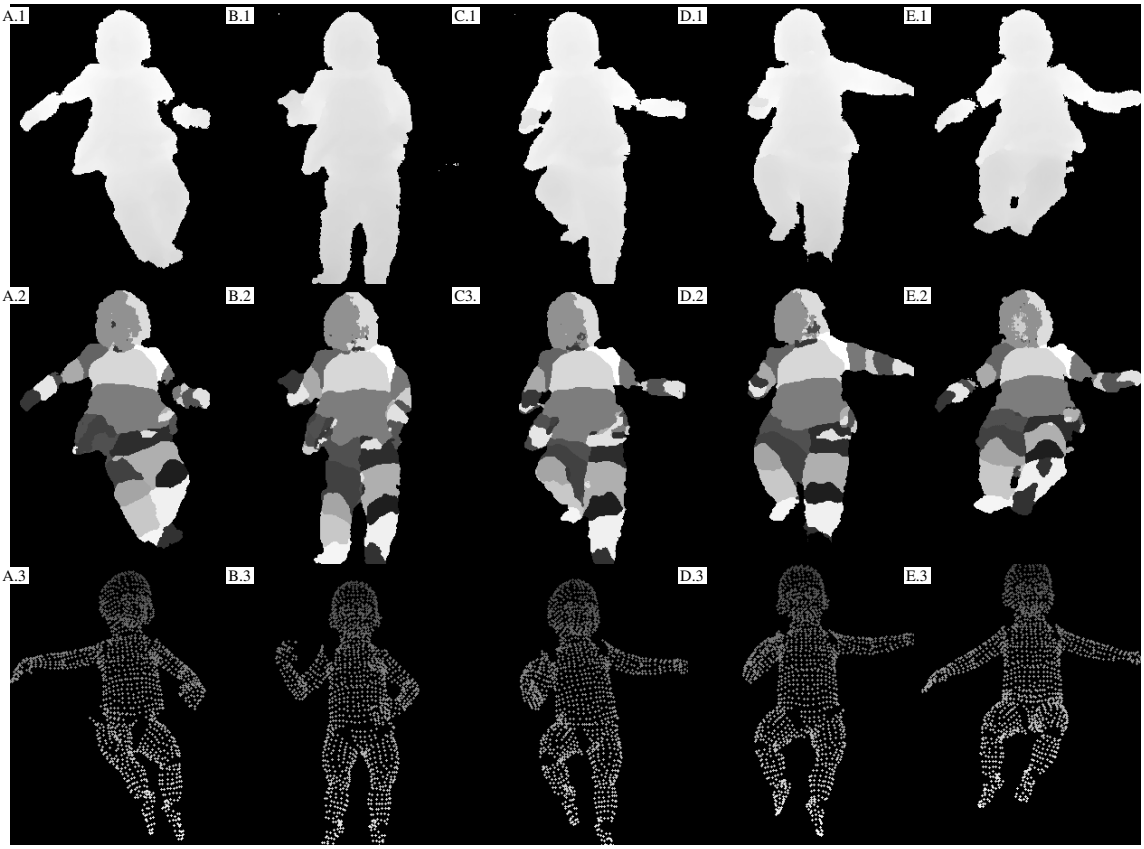

Figure 4.17: Sample frames 142, 210, 463, 499 and 535 from a sequence with 897 frames. A) Range image, B) Classification results, c) Articulated mode fit demonstrated via projected view

4.5 Conclusion

With the evidence presented in [69, 70], a correlation between an infant's kicking patterns and their development has been recognized. Furthermore, numerous tests evaluating the motions and actions of infants have also been established under similar theory. Their main assumption is that the range, speed and response of an infant's actions can serve as predictors for their health and development. To that end, it is evident that a method to automatically and consistently estimate the pose of an infant can serve to evaluate their development. Furthermore, it can provide evidence for early intervention which can have a profound impact on the child's quality of life.

A methodology outlining Robust Articulated Point-Set Tracking (RAPTr) was presented in Section 4.3. Both the method of capture and the training set generation are described in detail, the latter of which incorporates the articulated model to create a synthetic set, capable of capturing enough evidence of the subject's appearance to train a limb detector. Next, the section demonstrates how to connect the output limb detector predictions in the model-fitting strategy by utilizing the same articulated model to achieve a prediction-driven optimization. Lastly, a protocol for generating both moment-based and shape-based descriptors for the model fit and the subject's limb prediction is provided as the final function to estimate a better pose.

The findings presented, demonstrate the proposed approach is a viable means to estimate an infant's pose using a consumer grade range camera. With an average error of 1.78 cm, RAPTr outperforms the baseline method. Furthermore, as it achieved similar performance regardless of the limb detection method used, it demonstrates a robustness to classification error. Although false positives at the subject's extremities can prove to be challenging, on average the method does well. Furthermore, as frame independence is held as a hard constraint of this work, error does not propagate, implying that those bad frames have no impact on the sequence's remainder. Possible future works can focus on pursuing either faulty frame detection or tailored filtering methods to manage these specific error-producing conditions, while not impacting the well-fit frames.

CHAPTER 5
ROBUST ARTICULATED POINT-SET TRACKING (RAPTR): AN APPLICATION TO HUMAN POSE ESTIMATION

5.1 Introduction

Humans are the most common subjects for pose estimation problems. While there are example publications for pose estimation solutions designed for robots [101], animals [102, 103] and infants [38, 78, 84]. Most of the computer vision research community focuses their efforts on the estimation of adult human subjects [4, 5]. Applications that benefit from such solutions include but are not limited to security, gaming, and entertainment to name a few. Without a strong requirement for accuracy, rough estimation with real-time pose estimation is sufficient to meet the constraints for these applications.

Medical and physical therapy applications, on the other hand, require exceptional accuracy. Whether the topic is infant kicking analysis or clinical gait analysis, accuracy and repeatability is vital to achieving consistent and reliable results. As a physician's diagnosis can have a profound impact on the subject's medical outcome and essentially their quality of life, the evidence used to establish any recommendation must come from a solution with a low error margin. For this reason, some solutions in pose estimation drop the real time requirement and focus on adding additional methods, at the price of extra computational time, to achieve more accurate results [104]. To that end, the pose estimation problem space is made up of solutions that essentially make trade-offs in speed for accuracy, or vice versa.

In this chapter, the focus revolves on the use of articulated models (Figure 5.1) as both tools to estimate pose and as shape descriptor generators to train pose estimation regressors. Furthermore, the machine learning methods employed in this chapter are created using

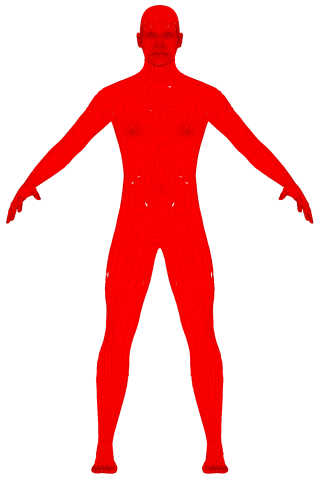

Figure 5.1: Makehuman adult male mesh model

the same articulated model tasked in estimating the subject's pose. Although there are plenty of works that utilize quicker algorithms to achieve pose estimation in real-time, the methods presented in this work return more accurate results at the cost of additional computation, outperforming recent works in accuracy. Furthermore, no effort was made to pursue real-time results. With adequate code optimization and exploitation of modern computing devices [105] this can be accomplished.

The solution presented in this work, utilizes the RAPTr algorithm presented in Chapter 4 on full humans. Provided with multiple subject capture samples from different range cameras, located at even intervals around the subject, an articulated point set model is fitted to the subject to estimate their pose (Figure 5.2.A-D) . Within this framework, as opposed to the prior work's use of only the subject's shape-based descriptors, both the moments from the detected subject and model fit are treated as feature descriptors. Additionally, the effective pose parameters established during fitting (both joint angle and joint position values) are treated as shape descriptors. With the resulting descriptor from their concatenation, a model is trained to map these features to their corresponding pose. Essentially, the estimate joint positions serve as a lower dimension approximation of the actual subject's

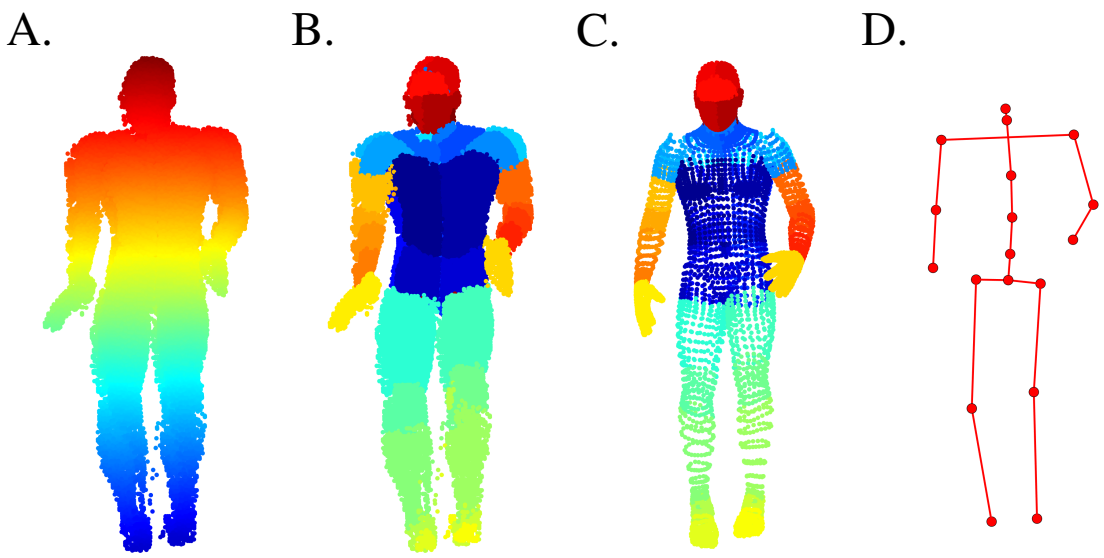

Figure 5.2: A) Point cloud capture, B) Classified point cloud, C) Model fit, D) Final estimate pose

pose.

The chapter is organized as follows. First, a related works section sets up a foundation for prior publications in the field, demonstrating recent efforts in 3d human pose estimation using point clouds and range images. Next, the methodology lays out the framework for the conversion of multiple views into a single point cloud, the limb detectors training, their use and inclusion in the model fitting strategy, the generation of the moment and shape descriptors, and lastly the training and use of the final regressor to return a final pose estimate in the pose space defined by the dataset. Next, the results section explores various combinations of the moments and shape descriptors, demonstrating which is the most effective when compared to the prior work's approach. Lastly, the conclusion summarizes the findings and provides a possible avenue for future work.

5.2 Related Works

Human pose estimation has been studied heavily in the last few decades [4]. Focusing solely on the 3d human pose estimation [5], numerous publications provide guidance on the baseline protocol. On average, the steps go as follows. First, ensure that the capture data from either range, stereo or mixed set of cameras is calibrated. Next, apply said calibration parameters and corresponding conversion method to the captured data to produce either a range-based or a point cloud-based representation of the subject. At this point, depending on the protocol, the output data is processed, and the pose is estimated.

Implementations for pose estimation on range or point cloud data follow a trend like traditional pose estimation. The model can be graph based or fully articulated. The mechanism for fitting or solving the pose can be holistic [106] (using a single algorithm to return entire pose), apply a unique representation which when optimized returns the pose [105], or bottom-up [28] (solving for the subject's limbs and then using those as inputs for a final pose estimate).

Holistic approaches have become a viable option with recent advancements in machine learning. In [22], the authors employ a pair of randomized decision forests. The first returns a per-pixel (in range image) classification, associating each pixel with their most probable limb. The second utilizes a regression tree model to learn the offsets needed to compensate for the fact that range cameras are only able to get the surface of a subject. The work presented in [107], uses a convolutional neural network that outputs a direct estimate of the pose parameters when provided an input range image. Also, efforts have been made towards mapping 2d images into 3d poses. In a similar fashion, the authors of [108] return a 3d pose for an input 2d image. In [109], the authors use a randomized decision forest to learn a local gradient that when applied to a point designed to do a random walk along the surface of a range image, will guide the point to convergence at the landmark of interest. A random forest is defined for each limb, thus the subject's pose is estimated when all the

random forest have been used and each corresponding random walk has converged.

Alternative representations have been demonstrated to be effective means to estimate 3d poses of subjects. Sums of Gaussian models have been demonstrated to have the capability to achieve real-time pose estimation with mixtures in both range [110] and multi-view monocular imagery [105]. Each limb is made up of a series of rigidly connected Gaussian functions, like what is presented in Chapter 4, with the weights defined by the average color for that corresponding limb. The method can estimate a series of complex poses from different views. However, it has temporal dependence and a manual initial definition of the model to match the subject as assumptions.

Bottom-up approaches were some of the first incarnations for 3d pose estimation solutions from range imagery. In [22], the authors employ a randomized decision forest to predict quick estimates of the subject's pose. They can achieve real-time results, trained only on synthetically generated humans. Also, the formulation admits frame independence from the prior or future frame, implying that error doesn't propagate. However, there are mixed results regarding accuracy for clinical purposes [111]. Furthermore, the method is limited as it is designed for subjects facing the camera and the poses used to generate the samples were captured with mocap data, implying a heavy upfront cost to data collection.

Recent developments in convolutional neural networks have provided powerful means to estimate 3d poses from 2d imagery. In [8], the authors can take the outputs from the hourglass [12] detector, process them using a full-connected network and gain a rough estimate of a subject 3d pose from 2d images, which can be in both outdoor and indoor settings.

In this study, the subject's pose is estimated from a series of captures taken simultaneously from cameras around the subject. In [7], the authors train a deconvolutional neural network to do semantic segmentation of the subject's limbs from multiple range image views. They apply a difficulty-based curriculum for training the network, beginning the training with a set of easy poses and ending with a final batch of difficult poses. The al-

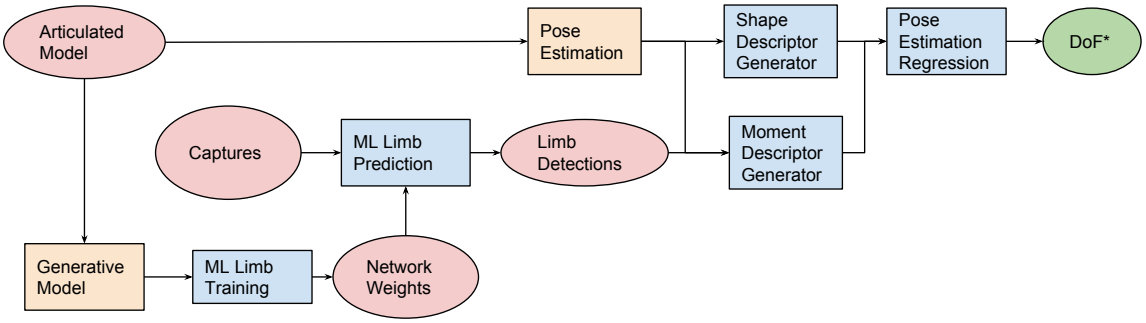

Figure 5.3: High level flow-chart of the proposed RAPTr Framework

gorithm can learn a series of poses ranging from standard upright poses to complex ones from different views. This is a requirement as their setup includes captures from three cameras evenly separated along a radial boundary of an area with the subject located at its center. A labeled point cloud is generated from the Multi-view predictions. They utilize a linear model trained on a per-class moment-based feature set to estimate the subject's final pose. The work presented in this chapter is designed to include their limb detector in the development of RAPTr to compare to their work and extend the multi-view pose estimation framework they have provided.

5.3 Methodology

This section describes a variant of the proposed work presented in Chapter 4 designed towards its use for full human pose estimation. Provided multiple paired range image captures of the subject, a dense point cloud representation is created (5.2.A). Next a limb detector processes the range images to create a per-limb class point cloud (5.2.B). An articulated model is then fitted to the point cloud using RPSR (5.2.C). Once the articulated model has converged to an optimal fit, a dense feature set is extracted from both the articulated model and subject and processed by a collection of linear regressors to get at the subject's pose estimate (5.2.D). Each step of the proposed framework is outlined in detail in the following section. A flowchart of the proposed approach is demonstrated in Figure

5.3.

Although there is a lot of overlap between this chapter and the study on full infant pose estimation from Chapter 4, there are number of differences that merit coverage. First, in the infant case there is only one camera, while in this problem there is a collection of cameras. Thus, a method to gain a consensus between the captured views is required. Next, with more input captures, a denser point cloud is produced. To that end, a down sampling approach is necessary to manage the bigger point cloud set. Also, the subject may be facing any given direction and with no floor sitting behind them, a wider range of poses need to be accounted for, requiring an alternative initialization strategy to be employed. Lastly, this work will explore using combining moment-based features to further inform the linear regression models trained for pose estimation and the gains in accuracy when doing so.

5.3.1 Datasets

In this study, the UBC Berkeley [7] and the MHAD [112] actions datasets are used. These sets provide a series of human poses with well over a million samples and their ground truth joint 3D locations. The number of points used for annotation differs per set, with the UBC set having 18 points while the MHAD set has 34. A skeletal representation of each set's ground truth points is demonstrated in Figure 5.11.A-B. The numbers designated on the skeleton are also the ids used to denote the per-limb error in the results section.

The UBC dataset is a collection of synthetic human subject images undergoing different poses, ranging from standing to laying down with many variations in between. The number of pose variations in the UBC set exceeded that of the MHAD set. Each capture in the UBC set is of a single human subject undergoing a completely random pose. Based on their surface model, the subject is a 1.6m tall, thin male. The set has been separated into three groups: easy, intermediate and hard. These levels of complexity are based on how far the poses deviate from the standing upright pose. Also, with three cameras capturing the subject, three sets of noiseless range and label image pairs are provided. Lastly, no forward-

Figure 5.4: A) Range image capture from behind the subject, B) Range image capture from in front of the subject, C) Resultant point cloud from their union

facing direction is designated. Thus, the subject may be directed towards any given camera. This is accounted for in the initialization methods.

A collection of human captures doing some predetermined actions makes up the MHAD action dataset. In this set, there are a total of 12 human subjects of different heights, sexes and shapes. Each subject was asked to undergo 11 specific actions like jumping or sitting, with each action being repeated 5 times. Since the actions are restricted to a finite set, the dataset holds less variation in poses than the UBC set. Also, only two cameras are included in this set, capturing naturally produced range images from the front and back of the subject. Evidence suggests that the subjects were instructed to face the same direction for every capture.

5.3.2 Calibration and Preprocessing

To estimate the 3D pose of the subject, a 3D point cloud must be generated with its corresponding per-point limb predictions. Doing so requires calibrating the range image cameras used to produce the capture to a fixed reference frame. Hence, both the extrinsic and intrinsic projection parameters need to be provided per camera for each capture. Furthermore, working within the scope of this study, the subject must be extracted from the capture. Essentially, this implies that the captures should be amenable to applying background subtraction or foreground segmentation to remove the point cloud points associated with the

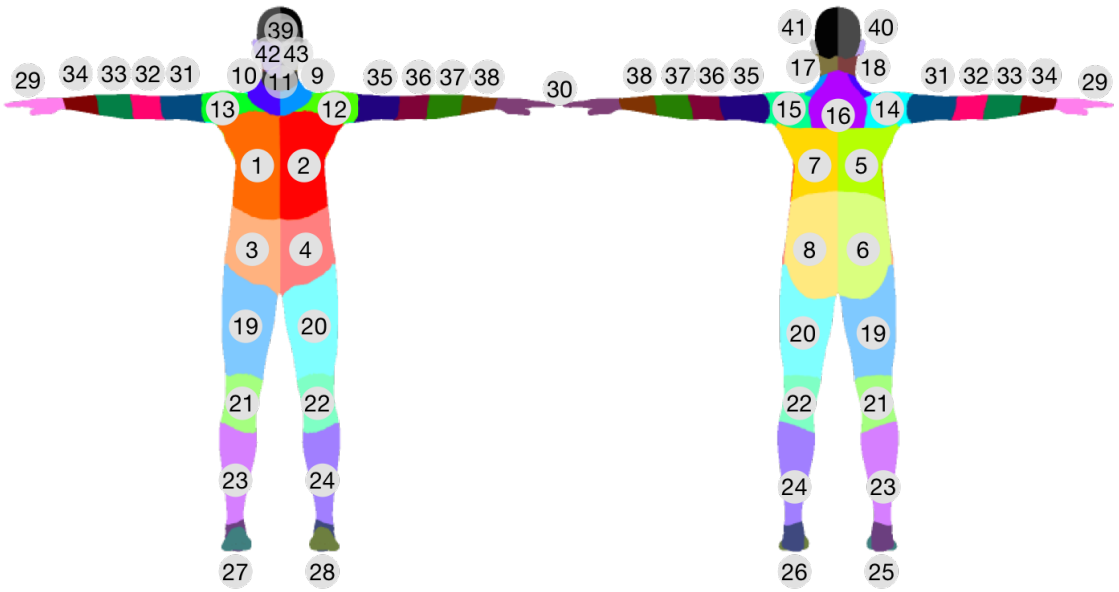

Figure 5.5: Numerical definition class map [7]

background or other objects in the space. Both sets used for testing had the calibration parameters and these qualities.

5.3.3 Adult Human Model

Calibration parameters are provided in both sets. For every camera included in the capture, there is a corresponding intrinsic and extrinsic parameter set. The extrinsic parameters are defined based on a common reference frame.

Converting the range image sets per capture into a single point cloud is done by using the already provided extrinsic parameters. Applying the inverse projection operator for each set using the calibration parameters provided, maps each range image pixel to its corresponding point in the subject's point cloud capture (Figure 5.4). The resultant point cloud is dense, the units are in meters and defined with respect to the same base frame as the camera models.

Additional steps are required for the MHAD dataset. As the set is comprised of real captures, a background subtraction step is needed to remove any points associated with the

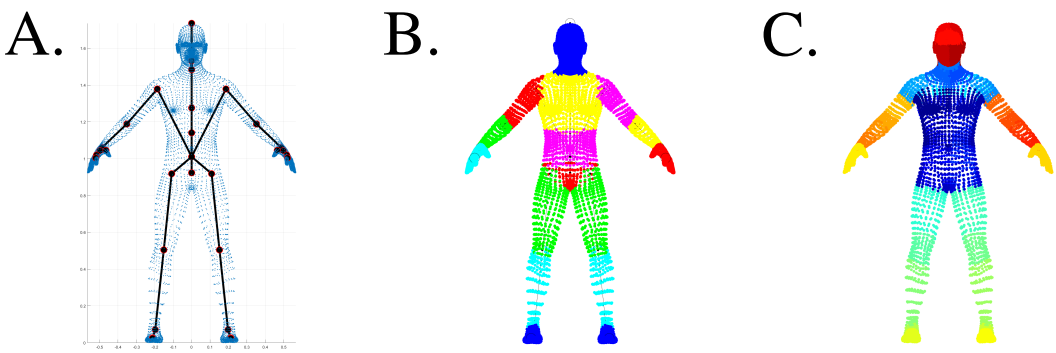

Figure 5.6: A) Point cloud and skeletal frame, B) Per-limb annotation C) Per-class annotation sample of the articulated model

floor or clutter in the space. An assumption of this work is that the input point cloud processed for pose estimation only includes subject points. Exploiting the geometry allows for simple thresholding of these values. Given the subject's position, a radius threshold applied using the subject's projected (x, y) center removes any points associated with clutter. Next, the floor points are detected by applying a RANSAC-based planar model (same one used in 4.3.2). A height threshold is set to one standard deviation of the measured noise from the planar model. Most of the floor points are removed. These two steps extract the full human subject.

An articulated point set model is provided by the Makehuman system. The articulated model's surface is defined by the collection of points, normals and colors. Its appearance is that of a fit, human male with a height of 1.6m. A skeletal frame is also provided, with the surface points rigidly assigned to their corresponding skeletal link. The color values applied to the articulated model are those used in the UBC label set class values as seen in Figure 5.5. They are assigned the same numerical value to define their corresponding label as per the definitions specified in [7]. Although a mesh model is provided, it is not used in this study.

In total, the model has 38 DoFs. That includes the shape parameters which are made up of 3 DoFs at each shoulder, hip-joint, ankle, wrist, neck and hip-to-back joint and 1 DoF at

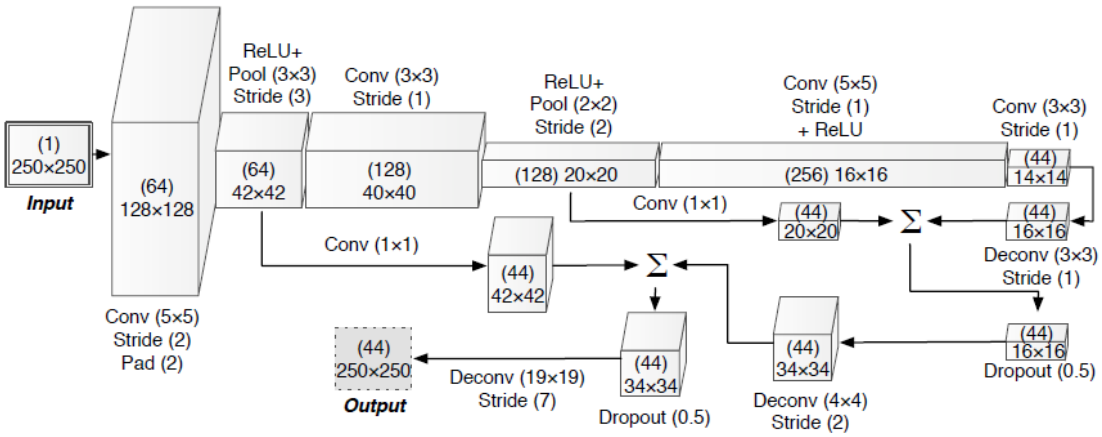

Figure 5.7: Semantic segmentation model employed in this study [7]

each elbow and knee joint. It also includes the six DoFs which control the 3d position and orientation of the subject are included.

An example of the articulated model and its skeleton is present in Figure 5.6.A, with a per-limb annotated articulated model in Figure 5.6.B and a per-class annotated articulated model in Figure 5.6.C.

5.3.4 Limb Detection using Semantic Segmentation

The per-pixel limb classifier from [7] is used as a limb detector in this work. The authors have made their implementation and saved model weights publicly available, hence no training is required. The author defines 43 class labels, each representing an individual section of the subject's surface. These classes are demonstrated in Figure 5.5, with a numerical value designated to them.

Although, architecturally speaking, the network differs greatly from the Segnet model employed in the prior chapter, it too returns per-pixel classification results. Serving as a deconvolutional neural network, the model uses several convolutions, max-pooling, stride inference operation modules and select skip connections. These operators feed into a final soft-max operator returning a the per-pixel classification prediction. For more details on the network architecture, see Figure 5.7.

Normalization is applied to each range image. At input, every image is cropped based on an estimated extent. The range of each image is defined by the maximum and minimum coordinate ranges that hold a non-zero-pixel value. These images are then resized, warping them to meet the predefined input image dimension used during training. In this application, the images were resized into 256x256 images. Next, the mask of non-zero valued pixels is extracted. After the input images are processed, this mask is applied to extract only the pixels associated with measured range values. As the method is a deconvolutional neural network, false positive predictions can bleed into the empty background. Thus, a step involving a simple mask operator product is included to remove these artifacts. Lastly, once the prediction is complete and the mask operation is applied, the image is then resized and zero padded to match its original shape and appearance at input time.

The output predictions are mapped to their corresponding point cloud points. The projective geometry defined in section 5.3.2 creates a one-to-one mapping between the pixels and their corresponding points in the capture point cloud. Hence, these prediction labels are naturally assigned to their place in the point cloud.

5.3.5 Model Initialization

Initialization of the articulated model is done based on the prediction results. Using the numerical values from Figure 5.5 as a reference, the articulated model's position group terms are set equal to the expectation of the torso label points [1-16]. This places the articulated model's torso at roughly the same place where the subject's torso is located. Next, the articulated model's group orientation is defined using a similar approach to the one employed in Section 4.3.6 . Here, the directional vectors are defined by the classes alone as there is no assumption on the subject's orientation with respect to the ground.

The initial conditions of the rotation matrix are defined as follows. The z vector is set equal to the direction of the torso-cluster center index 3, 4, 6, and 8 to the head-cluster center indexes 39-43. The x vector is the direction of the back-cluster center indexes 5-8

and 14-16 to the center of the front part of the torso indexes 1-4 and 9-13. Lastly, the y vector is defined by the right-side center indexes 1, 3, 5, 6, 10 and 13 to the left side center indexes 2, 4, 7, 8, 12 and 15. Once more, the Graham Schmidt transform is applied to the constructed rotation matrix to ensure it respects the orthonormal constraints.

Multiple label clusters are required for establishing the articulated model's initial conditions. Mostly torso labels are selected as they are the regions that have the least false positive rates. However, occlusion is a very common occurrence in these types of capture sets. Although multiple cameras are present, occlusion can still occur on certain parts of the body. This is especially true for the torso. It can be occluded by the subject's arms, head or legs. Thus, many cluster sets are used to define the initial group components for the subject. No additional parameters were estimated for initializing the articulated model.

Once initialized, the articulated model is updated based on Equation 5.5 until convergence.

5.3.6 Model Fitting

Once again, treating each point on the articulated model as a Gaussian function and having them rigidly assigned to a link within a fully articulated adult articulated model, allows for the use of the formulation presented in chapter 3, with f and h denoting the articulated model (Equation 3.1) and subject points (Equation 3.2), respectively. A function modeling the product of two mixtures of Gaussians is defined by

$$E(\theta) = \sum_{k=1}^{N_g} \sum_{i=1}^{N_k} \sum_{j=1}^{N_S} \hat{\alpha} \phi(x, \hat{\mu}_{ijk}, \hat{\Sigma}_{ijk}) \tag{5.1}$$

$$\hat{\mu}_{ijk} = g_k \mu_i - \nu_j \tag{5.2}$$

$$\hat{\Sigma}_{ijk} = R_k \Sigma_i R_k^T + \Gamma_j \tag{5.3}$$

$$\hat{\alpha} = \alpha_{i,k} \beta_j, \tag{5.4}$$

with g_k and R_k being the link's group function and rotation matrix, respectively. As the product of two Gaussians is also a Gaussian and differentiable, the derivative returns an explicit representation for the update.

The gradient update is

$$\frac{dE}{d\theta}(\theta) = \sum_{k=1}^{N_g}\sum_{i=1}^{N_k}\sum_{j=1}^{N_S}\hat{\alpha}\phi(x,\hat{\mu}_{ijk},\hat{\Sigma}_{ijk})\hat{\mu}_{ijk}^{T}\hat{\Sigma}_{ijk}^{-1}\frac{\delta A_k}{\delta\theta}J_k \quad , \tag{5.5}$$

with $\frac{\delta A_k}{\delta\theta}$ denoting the link twist derivative and J_k the Manipulator Jacobian. This derivative defines a direction in the pose parameters space, that is followed to retrieve an estimate for the subject's pose. The alpha value present in the Equation 5.5 is defined by the label correspondence between the subject and articulated model for the given point. Although this formulation was originally defined for single kinematic chain, it applies to an articulated object made up of multiple kinematic chains as well.

Using the gradient descent approach, the articulated model's pose in the gradient direction until it matches the subject's. By applying Equation 5.5 to the articulated model's DoF variables until convergence, an estimate of the pose is achieved. An articulated model is assumed to have converged when the update gradient's magnitude is less than a predefined value.

Alpha values are assigned to the articulated model's point in a natural way. The formulation in Equation 5.5 includes an α coefficient which can be designed to achieve better convergence. For this application, α is substituted by:

$$\alpha_{i,j} = \begin{cases} 1, & \text{if } C_i = C_j \\ 0, & \text{otherwise} \end{cases}, \tag{5.6}$$

with C_i and C_j representing the articulated model and subject point predicted class label, respectively. The resultant gradients end up directing the limbs to their corresponding sections in the subject point cloud. Additionally, as the "soft assignment" term α is in-

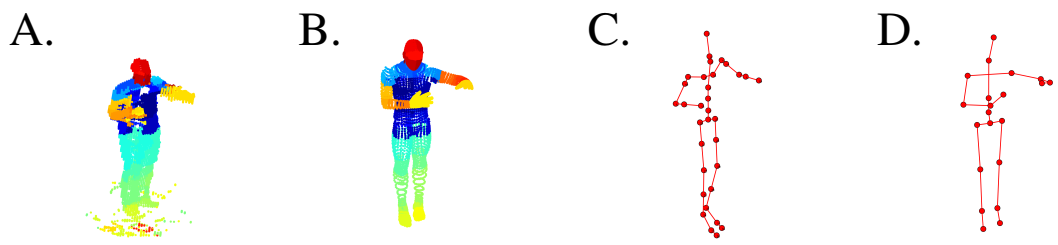

Figure 5.8: A) MHAD classified sample, B) Model fit of that sample, C) Truth Skeleton, D) Model Skeleton

cluded in the Equation above, a mechanism for directed fitting is inherited. Allowing the articulated model, the capability to converge to the predicted point cloud set with a wider region of attraction. Furthermore, the derivation has a robustness to both noisy points and false-positive classifications due to the predictions and Gaussian formulation, respectively.

Downsampling the subject point cloud is applied using a K-means approach. A fixed number of points are designated per label cluster to represent that section. The values for these points are updated using a K-means approach, converging to the set of points that best cover the span of their respective label data. Thus, a dense set located at the subject's head, enumerating more than $1000+$ points, is replaced with a collection of k representative Gaussian model centers. They do an adequate job at representing their corresponding limb. Also, each point is given equal weight to prevent an imbalance in the class representation. For this work, $k = 10$ demonstrated adequate performance.

5.3.7 Pose Estimation: Feature Generation

A collection of linear regressors is trained to estimate the subject's pose. As the position of a single point is made up of three values (x, y and z), three linear regression models are required for each point. Also, because each dataset has a different number of points, they will require a different number of regressions models to be trained. In the case of UBC, this implies 54 models. MHAD, on the other hand, requires 99 models. Each linear model is trained with the same input descriptor; however, they are trained to map to a different

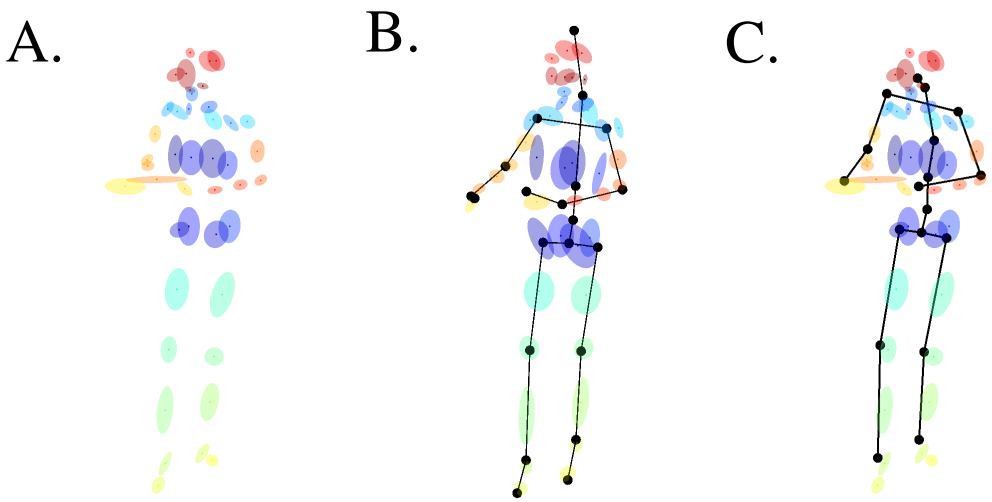

Figure 5.9: A) Moment-based descriptors from the subject, B) Moment-based descriptors from the model, superimposed on the model's skeleton, C) the ground truth skeleton

value.

Moment-based descriptors are extracted from each sample and fed to train the pose estimation linear regression model. Following the protocol defined in [7], the moment-based descriptors are extracted for each label group. A depiction of the moments is presented in Figure 5.9.A, with the shape of each ellipsoid representing one standard deviation. Also, each ellipsoid is colored based on the class they represent. The descriptor is made up of the mean, standard deviation, covariance, minimum value, maximum value and the eigenvalues of the covariance matrix. Redundant terms, like the repeated off diagonal values of the covariance matrix are excluded. Furthermore, these moment-based descriptors are also extracted from the articulated model as well (Ellipsoid in Figure 5.9.B). Once more, if no point for a given cluster are present, a zero vector of appropriate dimension is used as the feature descriptor for that class.

The articulated model, once fitted to the subject, serves as a shape feature generator. As the articulated model's shape mimics the subject's shape once it is fit, its DoF parameters serve as a lower dimensional interpretation of that fit. Thus, they can be treated as shape descriptors. Additionally, the effective positions of the articulated model link's end-

points (represented by the black markers in Figure 5.9.B) also mimic the subject and are also included as shape descriptors. In total, the DoF parameters are the model's position, orientation and joint angles.

Both the moment-based and shape-based descriptors are used to train the regression models. Taking the two sets of descriptors and concatenating them produces a dense feature descriptor. This descriptor is then used to train each regressor, returning a better fit tailored towards its respective dataset (Figure 5.9.C). The performance of this approach is analyzed in the following section.

5.4 Results and Discussion

In this section, the results and a discussion of their significance is presented. A goal of this work is to demonstrate the benefits of including the articulated model in the numerous steps involved in solving for a subject's pose. This includes using the articulated model to create the training set, fit with during the pose estimation step and finally treat as a feature generator to construct descriptors that are designed to train a refined linear regression model that estimates the subject's pose. As both the articulated model, the semantic segmentation model and the linear regression model are provided by [7], the only variable terms are the features used to train the linear regression model. It is their effectiveness that is being evaluated.

Application of the algorithms to three publicly available human pose estimation datasets serves to evaluate their effectiveness and validate the hypothesized benefits. The first two sets are from the UBC dataset. They consist of synthetic human range images with the samples designated as belonging to easy, intermediate or hard subsets. For this work, only the easy and hard datasets are tested on. The third dataset used in this work is the MHAD action dataset. The sample imagery is of real subjects undergoing selects actions. Evaluation of each method, including the baseline method, involves comparing the processed outputs to each set's corresponding ground truth values. An L2 error analysis between the

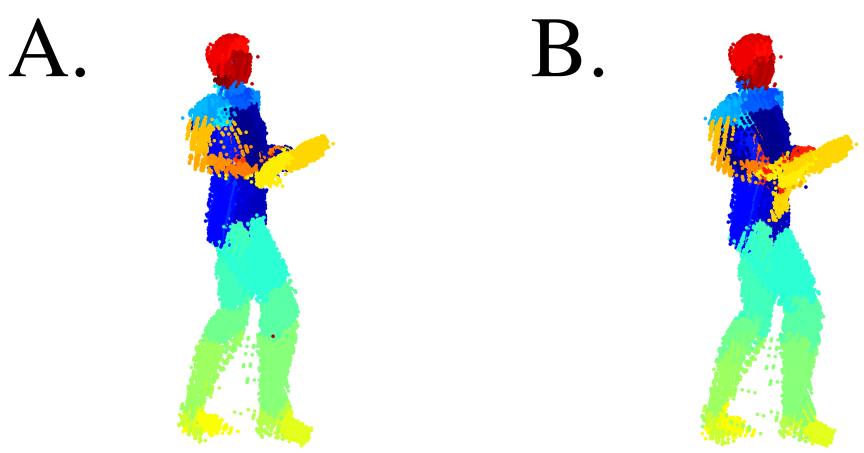

Figure 5.10: A) Sample from the GT Hard set, B) Sample from the predicted Hard set. In the prediction image, one can notice the labels bleeding into other regions while on the GT point cloud the labels are correctly placed. Specifically, error is noticeable in the hands and torso.

predicted and ground truth joint position quantifies performance. Further analysis includes the L2 per-limb error.

5.4.1 Datasets

There is a total of four subsets that are tested on from the UBC dataset. To demonstrate the effect of false positives, two versions of the easy and hard UBC datasets are used in this experiment. These are referred to as the ground truth (GT) and predicted version of the easy and hard UBC datasets. The ground truth set tests the system's capability to estimate the subject's pose when perfect per-limb classification is available. The prediction sets test how the method performs when false positives occur. In Figure 5.10 both a GT point cloud (A) and the predicted version of that point cloud (B) are demonstrated. A noticeable number of false positive classifications are apparent. For example, it is noticeable that the limb detector confused a a few labels between the torso and hand classes.

The MHAD actions dataset is comprised of captures of real adult human subjects and is tested on for evaluating the proposed method's performance. All the samples are prepro-

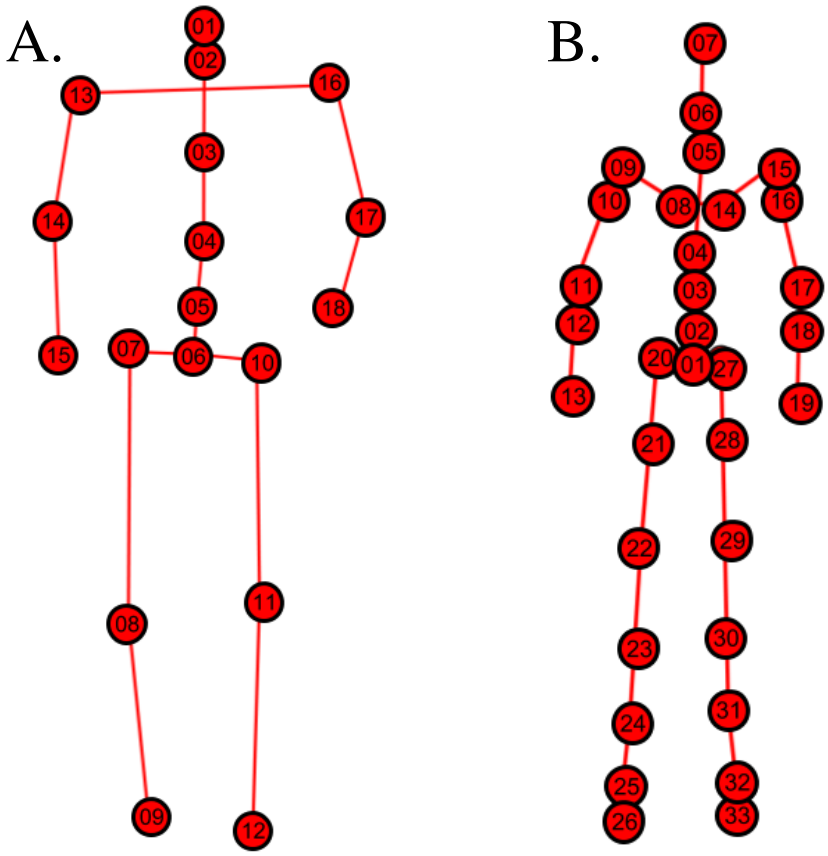

Figure 5.11: A) Skeletal frame from UBC dataset, B) Skeletal frame from MHAD

cessed using the protocol defined in Section 5.3.2.

All the observation sets provided in the UBC and MHAD datasets come with their corresponding pose estimate ground truth. In the pose estimation problem, the ground truth is made up of the true 3D joint positions. The UBC set has 18 marker positions, while the MHAD has 33. These 3D locations serve as both entries in the linear regression model training set and ground truth to measure the proposed work's accuracy. Figure 5.11.A-B demonstrates the skeletal frames for the UBC and MHAD datasets, respectively. These numerical indexes are references in the per-joint L2 position error plots below.

5.4.2 L2 Error Analysis

To evaluate the pose estimation performance of the proposed work, an L2 error analysis is applied. The articulated model presented in section 5.3.3 is fit to each point cloud based on

Table 5.1: Key for experiments

Reference Letter	Meaning
R	Real subject moment-based descriptors
B	Both subject and articulated model moment-based descriptors
A	DoF parameters
E	Articulated Model End-points

the proposed fitting function of the RAPTr framework. With the predictions defining a ”soft assignment”, the model converges to the point cloud, minimizing the differences between the limb correspondences. From this a series of feature descriptors, both moment-based and shape-based, are extracted for analysis. The generation of these features is described in Section 5.3.7. For the moment-based features, a feature set is estimated for each label point set.

Two sets of moment-based features are tested: the full set and a select set which is a subset of the full set. Employing a feature reduction protocol, the combinations of the output features in conjunction with the ground truth 3D joint positions are used to train a series of linear estimators. Each estimator predicts one value of the 3D pose set. The resultant error between the prediction and the ground truth is the indicator used to evaluate the proposed work's performance.

In this experiment, several combinations of the extracted features are evaluated. Ultimately the best feature combination is the one that when used to train the linear regression model, return the lowest L2 average error with a low variance. Both the moment-based and shape-based feature descriptors are estimated from the articulated model. From the subject point cloud, only the moment-based descriptors are estimated. In [7], the moment-based features are used to the train the linear estimators. This approach serves as a baseline for evaluation. In this work, however, an underlying assumption is that by training the regression models on the concatenated set of these features with a few additional features, derived from the fitted articulated model, improved pose estimation results can be achieved.

The linear regression models for each experiment are trained using the same protocol. Each regression model trained on its respective training set is trained on 75% of the data and tested on 25%. The tests are done 10 times each, with the sample indexes held constant per iteration to allow each combination of extracted feature to be tested on the same training samples. The outcomes of these runs are averaged on a per-joint basis.

To ease the readability of the numerous results which are a byproduct of the large number of combinations tested, a key has been created. In table 1, the key defining the shorthand mapping the features used is displayed. The source providing the features are denoted by "R" for just from the subject and "B" both from the subject and articulated model. "E" and "A" represent the fitted articulated model 's 3D joint positions and DoF, respectively.

Another test demonstrating the performance when all the label predictions vs when only those associated with areas around the joints are used to train the linear regression model is included in this study. A leading "F" is used to denote if all the labels are used and a leading "S" is used to denote if only the subset is used. For example, a test using the subset labels of both the subject moments plus the articulated model's endpoints and moments is referred to as "SBE".

Two types of charts are presented for each tested dataset. The first chart (Figures 5.12.A-E.1) shows the average L2 Error of all points for each combination of input features. The figure includes arrows representing one standard deviation and uses a color scheme to highlight specific outcomes. The colors are defined as follows: black, red and blue represent the baseline, the top performers and the rest, respectively. Additionally, every chart is paired with its corresponding per-joint error chart (Figures 5.12.A-E.2) . For the sake of display, only the top performing estimators (highlighted by red markers in the prior image) are presented. Also, to accommodate evaluations across the different dataset, the range of each chart are normalized to the range of the experiment which had the worst performance.

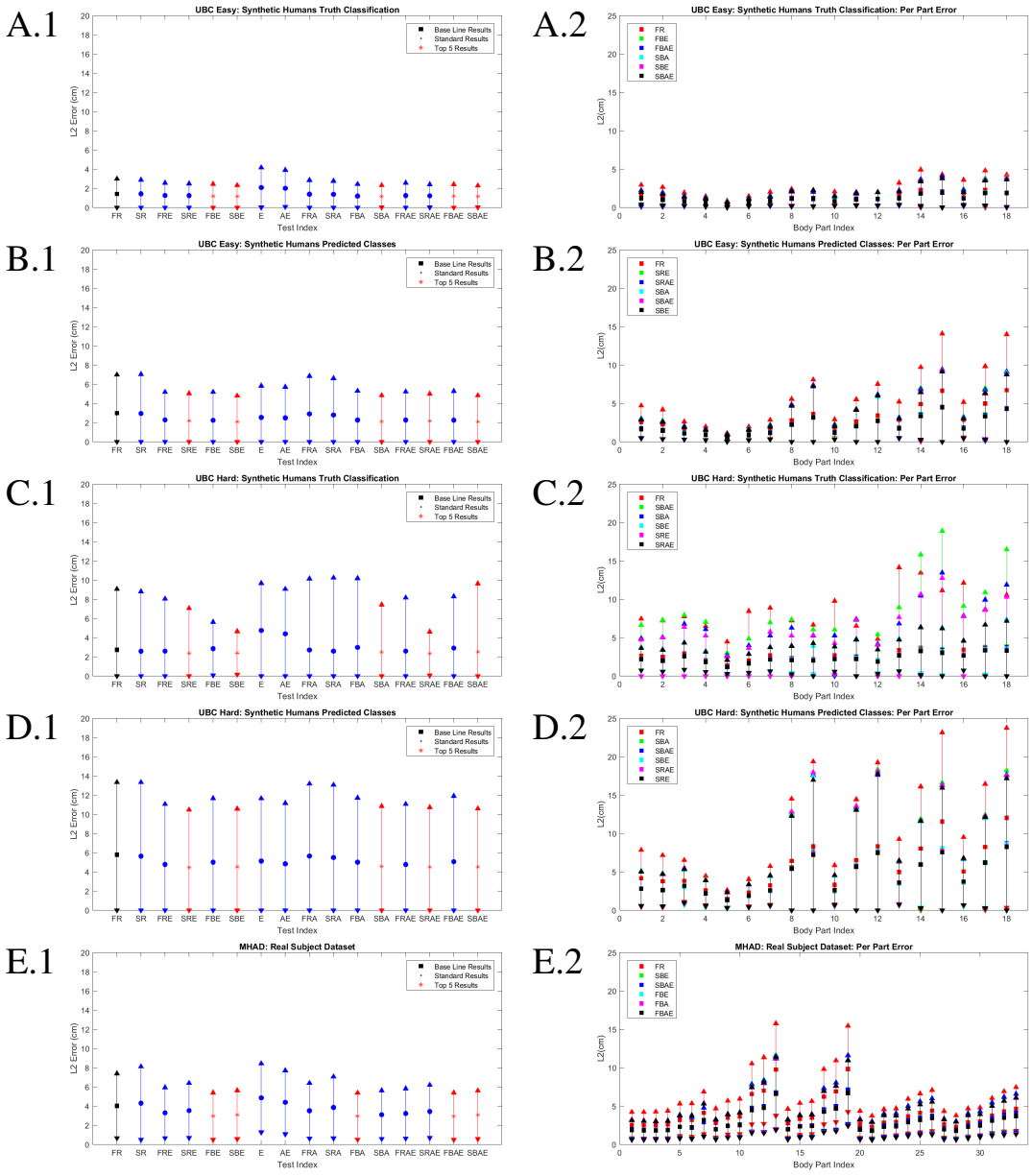

Figure 5.12: L2 Error Statistics. A-E.1) Per method error statistics, A-E.2) Per-part error statistics

5.4.3 Evaluating the Test Sets

UBC Easy and Hard Dataset: L2 Analysis

The UBC easy dataset returns the lowest error. Made up of mostly standing poses, the easy dataset has a large number of redundant poses which make it easier to for the linear regression model to learn the patterns and capture the underlying trends. As presented in Figures 5.12.1-2, the combined features outperform the baseline method with errors of 2.1 cm and 3.01 cm, respectively for the predicted set. This is a 30% improvement. The ground truth set, on the other hand, returns errors of 1.34 cm and 1.5 cm on average, respectively (14% lower error).

In the UBC Hard set, using the ground truth labels, the baseline denoted by "FR" returns an average error of 2.73 cm, while the top articulated model-based feature method "SRAE" obtains an error of 2.1 cm (23% lower error). A similar outcome is seen when using the predicted labels with the error for the competing method being 5.9 cm and the best performing articulated model-based method "SRAE" getting an error of 4.35 cm on average, 26.2% lower error. These results are presented in Figures 5.12.C.1 and 5.12.D.1, respectively.

MHAD: L2 Analysis

Evaluation of the MHAD dataset results shows that the articulated model's inclusion in the pose estimator's training returns improved results compared to the competing method (Figure 5.12.A.1) for real subject pose estimation. When the regression model is trained only on the moment-based features derived solely from the subject limb predictions, an average error of 4.1 cm is achieved. Using the same samples to generate the articulated model-based features (moment and shape descriptors), returns better results. In the "FBAE", for the real human subject set, an error of 3.4 cm is achieved, equivalent to 17% lower error. As displayed in Figure 5.12.E.2, this outcome also returns better per joint error with the best

method shown in black and the competing approach in red. The other top performing feature combinations are ”FBE”, ”SBE”, ”FBA” and ”SBAE”. All of which are combinations made up of features derived from the articulated model.

5.4.4 Discussion

Reviewing the results presented in Figures 5.12.A-E, there is a clear performance gain when using the RAPTr system. Incorporating the estimated articulated model features in the linear regression model's training returns 17% better accuracy on average. In each of the UBC and MHAD sets, the best errors are returned by the regressors trained on some combination that includes the articulated model features. The poorest results were seen when training the linear regression model on only shape-based descriptors from the articulated model. Training on only the DoF parameters resulted in 13 cm error on average for the MHAD set. These findings are not included in the figure as they would make the other method's results more difficult to see.

A noticeable pattern from this work is that more accurate pose estimates are achieved by including features that model the subject's shape. Ultimately, the aim of this study is to get better pose estimation error. At the cost of additional computation time, that has been accomplished as demonstrated by the evidence. One possible reason for the performance gain is that the articulated model approximates the subject's shape better than a graph-based model. With the articulated model including hard assumptions about the subject's geometry, dimensions and physiology, the optimal fit places the articulated model's joints closer to the true unobservable subject's joints. Also, since the articulated model is rigid, it is unable to ”blow up” due to very in accurate limb detections or noisy data.

Recently, the majority of publications in human pose estimation has focused on 2D-to-3D. In these works, the authors restrict their approaches to working with only 2D imagery. Each image has its corresponding 2D and 3D joint position annotations available. The bulk of these works, utilize the Human3.6M dataset as well. Although their problem sets are out-

Table 5.2: Error Analysis Comparison with Other Approaches on Real Subjects. The error presented is the average error and the units are in centimeters (cm).

Year	Method	Error	Dataset Name	Size	Capture Type
2010	Bo [113]	6.37	HumanEva [35]	80,000	Monocular
2014	Kostrikov [114]	11.57	HumanEva	80,000	Monocular camera
2015	Jung [115]	5	EVA [116]	-	Single range camera
2016	Yasin [117]	10.83	Human3.6M [24]	380,000	Monocular camera
2016	Pavlakos [118]	6.69	Human3.6M	380,000	Monocular camera
2016	Shafaei [7]	5.01	MHAD	85,000	Two range cameras
2017	Mehta [9]	8.05	Human3.6M	190,000	Monocular camera
2017	Martinez [108]	6.75	Human3.6M	190,000	Monocular camera
2018	Rogez [119]	8.81	Human3.6M	190,000	Monocular camera
2018	Rhodin [120]	14	Human3.6M	190,000	Monocular camera
2018	RAPTr	3.4	MHAD	17,000	Two Range cameras

side of the scope of this project, their relative error is comparable to the results presented in this chapter. With an increase in error of around 4cm on average, these approaches are capable of estimating poses of subjects from a single image captured from a monocular camera. With an average error of 6.75 cm, the authors of [108] are able to estimate the 3D pose from 2D joint detections. Furthermore, when using the ground-truth 2D detections, they are capable of obtaining an average error of 4.7 cm. In the Vnect [9] publication, auxiliary pose estimation tasks are imposed in the middle of the network. Focusing primarily on learning an alternative bone length map and initial joint locations, the network returns an in-frame joint location heat map and a set of finalized pose estimates. Demonstrating the benefits of utilizing auxiliary tasks to improve pose estimation, the method is capable of an 8.05 cm average error.

Lastly, a new line of investigation in pose estimation is the study towards the use of generalized adversarial networks (GANS) for the purpose of extending the utility of training sets and increasing the pose estimation accuracy. Two such examples [119] and [120] achieve errors of 8.81 cm and 14.56 cm average error, respectively. These works represent a Although these works report higher error, they still demonstrate possible avenues for

extending the methodology employed here. Namely, the benefit of utilizing an intermittent articulated model fitting method can return additional information that should improve their results. In particular, with these approaches serving to estimate model's initial conditions and assuming a viable interaction matrix is available through calibration, any of these methods can utilized in the RAPTr framework.

Comparison of the RAPTr results with other works is presented in Table 5.2. The table includes the datasets used and capture type. The proposed framework out performs the prior methods with similar capture set ups.

It should be noted that a large error variance presents itself at the extremities for all methods in the per joint error analysis (Figures 5.12.A-E.2). Namely, the largest error is present at the outer most joints (e.g. hands and feet). The indexes referenced in the error plots are plotted over their corresponding joints in Figure 5.11.A-B. The error present can be an indicator of the articulated model failing to fit to these specific points. A likely cause for this is the chained effect error can have on kinematic systems: for e.g. the error from the torso induces error on the shoulders which creates further error along the arm. Evidence suggests that the largest error is predominately located at the hands. However, inclusion of the articulated model-based features returns better on average joint position error when compared to the baseline, implying that the inclusion of these poor fits into the linear regression model's training set, allows it to learn from and compensate for said error. Also, although a lower variance is present for all approaches when the ground truth labels are used (Figure 5.12.A, C), a similar gain in performance is present when the regression model is trained on some articulated model features.

5.5 Conclusion

Accurate human pose estimation has already provided numerous benefits at the cost of forcing the subject to wear markers. Marker tracking software like the Viacom and Optitrack have aided multiple applications in physical therapy, medicine and motion pictures to

name a few. However, these systems come at a cost that is often too great for any clinic or person to afford. One common goal amongst pose estimation solutions, is to create solutions that can operate with consumer grade cameras, a group of devices including low cost depth cameras. In this chapter, such a solution was proposed that can provide an alternative means to give access to the benefits reliable pose estimation can make possible in a way that is both affordable and practical.

An outline of the proposed RAPTr framework was presented for use on human subject pose estimation. The method uses a deconvolutional neural network as a limb detector to process the input range images. These images, with the calibration parameters available, are combined to form a point cloud representation of the subject. The limb detections are then mapped into the corresponding points on the point cloud as well. Using both the predictions and the point cloud, a human articulated model is fit using the RPSR. Lastly, using features generated from both the articulated model's and the subject's point cloud, a linear estimator is trained to return an estimate of the subject's 3D joint positions.

The proposed approach, which merges the benefits of articulated model-driven and machine learning based methods, provides consistent pose estimation. As presented in the results section, the proposed methodology outperforms the baseline approach, implying that including an articulated model as a feature generator introduces additional information that results in better pose estimation. The only cost is the overhead of doing the articulated model fitting.

There are a few directions that this research can take. One, would be to explore methods to speed the up the articulated model fitting so that real-time results can be achieved while still returning accurate estimates. The second would be to test more complex regression models as an alternative to the linear regressor. Lastly, evaluate the viability and performance of this approach in a clinical environment to identify steps leading to practical real-world use in medicine.

CHAPTER 6
CONCLUSION

6.1 Conclusion

In this work, a method for human pose estimation called Robust Articulated Point Set Tracking (RAPTr) system is presented. The various system components are explored through a few physical therapy-based applications. First, a per-pixel limb detector is derived for estimating a series of clinical gait metrics for use with 2D videos. Next, an extension to the Robust Point Set Registration (RPSR) algorithm for tracking the kicking habits of an infant's single leg is created. The resultant pose estimates over time provides a means to read the joint signals, providing a glimpse into the subject's kicking patterns. Then, the definition of the RAPTR system which is a combination of the limb detector theory from the first application and the articulated model fitting theory from the second application is presented. Two applications are explored with the RAPTr system. The first is a full infant pose estimator from a single range image and a full human pose estimator given multiple range image views. This work's key contributions are summarized as follows:

- A 2D limb detector that processes silhouettes of humans walking and returns limb detections is applied as a method to estimate clinical gait metrics. Estimated from captures taken by a monocular camera, the detections estimate a few gait metrics which include step length, stride length, cadence and step speed. Additionally, an SVM model fit on a per frame basis to estimate the heel strike and toe off angles of a subject's walking gait.

- An extension of the RPSR formulation towards 3D articulated model fitting of an infant's single leg is presented. Additionally, a point-dependent coefficient is included in the formulation to define a weighted "soft assignment" between the model's and

subject's corresponding points. This assignment creates a larger region of attraction for the model's convergence to the subject's true pose.

- Definition of the RAPTr system for use as a full infant pose estimator from kinect range image captures. In this case, the coefficient definition is based on the limb detection predictions and their definition on the model used to generate the limb detector's training set. Furthermore, a procedure to use the model as a shape-based feature generator to train a linear regression to estimate the subject's pose.

- A modification of the RAPTr system for use in full human pose estimation on point clouds generated from multiple range image views. The same mapping approach used on the full infant pose estimator is applied. The study evaluates the use of both moment-based and shape-based feature descriptors from the articulated model for training the linear regression models used for the final subject's pose estimate.

The proposed approach has a number of limitations to its use. With the extension of Chapter 2's framework to allow for tracking of the infant's full body, now there is limit to how much interaction the adult can have with their child during the capture. Before, the adult could interact with the infant's upper body. However, now as occlusion must be kept to a minimum, they must refrain from covering any view of the child from the camera. Next, with the inclusion of an articulated model, a single subject can have their pose estimated. However, the system is not designed to handle multiple individuals, a possible extension to this research.

Future works that may be explored as extensions of this work include but are not limited to a method which can achieve real-time performance of the RAPTr system. The implementation presented is not designed for real-time practice. For use in a real world clinical setting, the algorithm must be optimized. Doing so will allow for incorporation of the system's use in an actual clinical case study. The benefits of this approach should be explored. Furthermore, instead of learning a mapping from the input feature space the output pose

space, another line of inquiry can be how to train a model to return clinically relevant metrics. With respect to the computer vision community, an exploration of the possible gains from including alternative model-based feature descriptors into the regressor set used to estimate the final pose may lead to more accurate gains.

REFERENCES

[1] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey,"

[2] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[3] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition.*, IEEE, 2008, pp. 1–8.

[4] X. Perez-Sala, S. Escalera, C. Angulo, and J. Gonzalez, "A survey on model based approaches for 2d and 3d visual human pose recovery," *Sensors*, vol. 14, no. 3, pp. 4189–4210, 2014.

[5] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1995–2006, 2013.

[6] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition.*, IEEE, 2009, pp. 1014–1021.

[7] A. Shafaei and J. J. Little, "Real-time human motion capture with multiple depth cameras," in *Computer and Robot Vision (CRV), 2016 13th Conference on*, IEEE, 2016, pp. 24–31.

[8] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching," in *CVPR*, vol. 2, 2017, p. 6.

[9] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 44, 2017.

[10] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014, pp. 1653–1660.

[11] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 1561–1570.

[12] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*, Springer, 2016, pp. 483–499.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[16] M. Eichner, V. Ferrari, and S Zurich, "Better appearance models for pictorial structures.," in *BMVC*, vol. 2, 2009, p. 5.

[17] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 623–630.

[18] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *IEEE Conference on Computer Vision and Pattern Recognition.*, IEEE, 2011, pp. 1385–1392.

[19] B. Sapp, A. Toshev, and B. Taskar, "Cascaded models for articulated pose estimation," *European Conference on Compute Vision*, pp. 406–420, 2010.

[20] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "2d articulated human pose estimation and retrieval in (almost) unconstrained still images," *International Journal of Computer Vision*, vol. 99, no. 2, pp. 190–214, 2012.

[21] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Body parts dependent joint regressors for human pose estimation in still images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2131–2143, 2014.

[22] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, *et al.*, "Efficient human pose estimation from single depth images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2821–2840, 2013.

[23] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[24] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[25] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele.

[26] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, "Personalizing human video pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[27] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[28] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[29] *Makehuman*, `http://www.makehumancommunity.org/`, Accessed: 2018-06-30.

[30] S. Johnson and M. Everingham, "Combining discriminative appearance and segmentation cues for articulated human pose estimation," in *International Conference on Computer Vision Workshops*, IEEE, 2009, pp. 405–412.

[31] H. Jiang, "Human pose estimation using consistent max covering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1911–1918, 2011.

[32] L. Ladicky, P. H. Torr, and A. Zisserman, "Human pose estimation using a joint pixel-wise and part-wise formulation," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 3578–3585.

[33] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 3041–3048.

[34] J. Bandouch, F. Engstler, and M. Beetz, "Evaluation of hierarchical sampling strategies in 3d human pose estimation.," in *BMVC*, 2008, pp. 1–10.

[35] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1-2, p. 4, 2010.

[36] V. John, E. Trucco, and S. Ivekovic, "Markerless human articulated tracking using hierarchical particle swarm optimisation," *Image and Vision Computing*, vol. 28, no. 11, pp. 1530–1547, 2010.

[37] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[38] M. M. Serrano, Y.-P. Chen, A. Howard, and P. A. Vela, "Lower limb pose estimation for monitoring the kicking patterns of infants," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, IEEE, 2016, pp. 2157–2160.

[39] C. Kirtley, *Clinical gait analysis: theory and practice*. Elsevier Health Sciences, 2006.

[40] M. Isken, T. Frenken, M. Frenken, and A. Hein, "Towards pervasive mobility assessments in clinical and domestic environments," in *Smart Health*, Springer, 2015, pp. 71–98.

[41] T. M. Gill, C. S. Williams, and M. E. Tinetti, "Assessing risk for the onset of functional dependence among older adults: The role of physical performance," *Journal of the American Geriatrics Society*, vol. 43, no. 6, pp. 603–609, 1995.

[42] M. G. Benedetti, V. Agostini, M. Knaflitz, V. Gasparroni, M. Boschi, and R. Piperno, "Self-reported gait unsteadiness in mildly impaired neurological patients: An objective assessment through statistical gait analysis," *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 1, 2012.

[43] R. W. Kressig, O. Beauchet, *et al.*, "Guidelines for clinical applications of spatio-temporal gait analysis in older adults," *Aging clinical and experimental research*, vol. 18, no. 2, pp. 174–176, 2006.

[44] M Stijntjes, C. Meskers, A. de Craen, R. van Lummel, S. Rispens, P. Slagboom, and A. Maier, "Effect of calendar age on physical performance: A comparison of standard clinical measures with instrumented measures in middle-aged to older adults," *Gait & Posture*, vol. 45, pp. 12–18, 2016.

[45] J. D. Ries, J. L. Echternach, L. Nof, and M. G. Blodgett, "Test-retest reliability and minimal detectable change scores for the timed "up & go" test, the six-minute walk test, and gait speed in people with alzheimer disease," *Physical therapy*, vol. 89, no. 6, pp. 569–579, 2009.

[46] M. B. van Iersel, W. Hoefsloot, M. Munneke, B. R. Bloem, and M. M. O. Rikkert, "Systematic review of quantitative clinical gait analysis in patients with dementia," *Zeitschrift für Gerontologie und Geriatrie*, vol. 37, no. 1, pp. 27–32, 2004.

[47] B. Jolles, A Grzesiak, A Eudier, H. Dejnabadi, C Voracek, C. Pichonnaz, K. Aminian, and E. Martin, "A randomised controlled clinical trial and gait analysis of fixed- and mobile-bearing total knee replacements with a five-year follow-up," *Journal of Bone & Joint Surgery, British Volume*, vol. 94, no. 5, pp. 648–655, 2012.

[48] W. A. Stuberg, V. L. Colerick, D. J. Blanke, and W. Bruce, "Comparison of a clinical gait analysis method using videography and temporal-distance measures with 16-mm cinematography," *Physical Therapy*, vol. 68, no. 8, pp. 1221–1225, 1988.

[49] H Stolze, J. Kuhtz-Buschbeck, C Mondwurf, A Boczek-Funcke, K Jöhnk, G Deuschl, and M Illert, "Gait analysis during treadmill and overground locomotion in children and adults," *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, vol. 105, no. 6, pp. 490–497, 1997.

[50] M. Gabel, R. Gilad-Bachrach, E. Renshaw, and A. Schuster, "Full body gait analysis with kinect," in *International Conference of Engineering in Medicine and Biology Society*, IEEE, 2012, pp. 1964–1967.

[51] A. Pfister, A. M. West, S. Bronner, and J. A. Noah, "Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis," *Journal of medical engineering & technology*, vol. 38, no. 5, pp. 274–280, 2014.

[52] E. E. Stone and M. Skubic, "Passive in-home measurement of stride-to-stride gait variability comparing vision and kinect sensing," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, IEEE, 2011, pp. 6491–6494.

[53] R. Z.-L. Hu, A. Hartfiel, J. Tung, A. Fakih, J. Hoey, and P. Poupart, "3d pose tracking of walker users' lower limb with a structured-light camera on a moving platform," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops.*, IEEE, 2011, pp. 29–36.

[54] V. Bonnet, C. A. Coste, L. Lapierre, J Cadic, P. Fraisse, R. Zapata, G Venture, and C. Geny, "Towards an affordable mobile analysis platform for pathological walking assessment," *Robotics and Autonomous Systems*, vol. 66, pp. 116–128, 2015.

[55] E. Surer, A. Cereatti, E. Grosso, and U. Della Croce, "A markerless estimation of the ankle–foot complex 2d kinematics during stance," *Gait & posture*, vol. 33, no. 4, pp. 532–537, 2011.

[56] A. Castelli, G. Paolini, A. Cereatti, and U. Della Croce, "A 2d markerless gait analysis methodology: Validation on healthy subjects," *Computational and Mathematical Methods in Medicine*, vol. 2015, 2015.

[57] J. Courtney and A. M. De Paor, "A monocular marker-free gait measurement system," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 4, pp. 453–460, 2010.

[58] J. Saboune and F. Charpillet, "Markerless human motion capture for gait analysis," *arXiv preprint cs/0510063*, 2005.

[59] E. Ribnick and N. Papanikolopoulos, "3d reconstruction of periodic motion from a single view," *International Journal of Computer Vision*, vol. 90, no. 1, pp. 28–44, 2010.

[60] J. Wang, H. Man, and Y. Yin, "Tracking human body by using particle filter gaussian process markov-switching model," in *International Conference on Pattern Recognition*, IEEE, 2008, pp. 1–4.

[61] M. Goffredo, J. N. Carter, and M. S. Nixon, "2d markerless gait analysis," in *European Conference of the International Federation for Medical and Biological Engineering*, Springer, 2009, pp. 67–71.

[62] A Leu, D Ristić-Durrant, *et al.*, "A robust markerless vision-based human gait analysis system," in *International Symposium on Applied Computational Intelligence and Informatics*, IEEE, 2011, pp. 415–420.

[63] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*, Springer, 2002, pp. 135–144.

[64] B. Sargent, H. Reimann, M. Kubo, and L. Fetters, "Quantifying learning in young infants: Tracking leg actions during a discovery-learning task," *Journal of Visualized Experiments*, no. 100, e52841–e52841, 2015.

[65] D. G. Stephen, W.-H. Hsu, D. Young, E. L. Saltzman, K. G. Holt, D. J. Newman, M. Weinberg, R. J. Wood, R. Nagpal, and E. C. Goldfield, "Multifractal fluctuations in joint angles during infant spontaneous kicking reveal multiplicativity-driven coordination," *Chaos, Solitons & Fractals*, vol. 45, no. 9, pp. 1201–1219, 2012.

[66] B. Sargent, N. Schweighofer, M. Kubo, and L. Fetters, "Infant exploratory learning: Influence on leg joint coordination," *PloS one*, vol. 9, no. 3, e91500, 2014.

[67] B. Sargent, J. Scholz, H. Reimann, M. Kubo, and L. Fetters, "Development of infant leg coordination: Exploiting passive torques," *Infant Behavior and Development*, vol. 40, pp. 108–121, 2015.

[68] C. Einspieler, H. F. Prechtl, F. Ferrari, G. Cioni, and A. F. Bos, "The qualitative assessment of general movements in preterm, term and young infants—review of the methodology," *Early human development*, vol. 50, no. 1, pp. 47–60, 1997.

[69] L. Fetters, Y.-p. Chen, J. Jonsdottir, and E. Z. Tronick, "Kicking coordination captures differences between full-term and premature infants with white matter disorder," *Human movement science*, vol. 22, no. 6, pp. 729–748, 2004.

[70] L. Fetters, I. Sapir, Y.-p. Chen, M. Kubo, and E. Tronick, "Spontaneous kicking in full-term and preterm infants with and without white matter disorder," *Developmental psychobiology*, vol. 52, no. 6, pp. 524–536, 2010.

[71] M. C. Allen, "Neurodevelopmental outcomes of preterm infants," *Current opinion in neurology*, vol. 21, no. 2, pp. 123–128, 2008.

[72] E. Blair and L. Watson, "Epidemiology of cerebral palsy," in *Seminars in Fetal and Neonatal Medicine*, Elsevier, vol. 11, 2006, pp. 117–125.

[73] M. Hadders-Algra, "Early diagnosis and early intervention in cerebral palsy," *Improving outcomes in cerebral palsy with early intervention: new translational approaches*, p. 9, 2015.

[74] L Meinecke, N Breitbach-Faller, C Bartz, R Damen, G Rau, and C Disselhorst-Klug, "Movement analysis in the early detection of newborns at risk for developing spasticity due to infantile cerebral palsy," *Human movement science*, vol. 25, no. 2, pp. 125–144, 2006.

[75] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures for multiple human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1669–1676.

[76] M. D. Olsen, A. Herskind, J. B. Nielsen, and R. R. Paulsen, "Body part tracking of infants," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, IEEE, 2014, pp. 2167–2172.

[77] N. Hesse, G. Stachowiak, T. Breuer, and M. Arens, "Estimating body pose of infants in depth images using random ferns," in *IEEE International Conference on Computer Vision Workshops*, 2015, pp. 35–43.

[78] N. Hesse, A. S. Schroeder, W. Müller-Felber, C. Bodensteiner, M. Arens, and U. G. Hofmann, "Markerless motion analysis for early detection of infantile movement disorders," in *EMBEC & NBC 2017*, Springer, 2017, pp. 197–200.

[79] N. B. Karayiannis, B. Varughese, G. Tao, J. D. Frost Jr, M. S. Wise, and E. M. Mizrahi, "Quantifying motion in video recordings of neonatal seizures by regularized optical flow methods," *Image Processing, IEEE Transactions on*, vol. 14, no. 7, pp. 890–903, 2005.

[80] A. Stahl, C. Schellewald, Ø. Stavdahl, O. M. Aamo, L. Adde, and H. Kirkerød, "An optical flow-based method to predict infantile cerebral palsy," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 20, no. 4, pp. 605–614, 2012.

[81] H. Rahmati, O. M. Aamo, O. Stavdahl, R. Dragon, and L. Adde, "Video-based early cerebral palsy prediction using motion segmentation," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, IEEE, 2014, pp. 3779–3783.

[82] S. Qin, Y. Nagai, A. Nakatani, H. Fukuyama, M. Myowa-Yamakoshi, and M. Asada, "Understanding information transfer in caregiver-infant interaction,"

[83] H Friedman, O Bar-Yosef, G Gordon, O Forkosh, and E Schneidman, "Analysis of infant neuromotor development using a computer-based approach," *Neural Plasticity and Cognitive Modifiability*, p. 35, 2013.

[84] M. D. Olsen, A. Herskind, J. B. Nielsen, and R. R. Paulsen, "Model-based motion tracking of infants," in *European Conference on Computer Vision*, Springer, 2014, pp. 673–685.

[85] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3d pose estimation from a single depth image," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 731–738.

[86] D. Kurmankhojayev, N. Hasler, and C. Theobalt, "Monocular pose capture with a depth camera using a sums-of-gaussians body model," in *Pattern Recognition*, Springer, 2013, pp. 415–424.

[87] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[88] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the conference on Visualization'02*, IEEE Computer Society, 2002, pp. 163–170.

[89] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 2389–2395.

[90] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, IEEE, 1991, pp. 2724–2729.

[91] S. S. Shivakumar, H. Loeb, D. K. Bogen, F. Shofer, P. Bryant, L. Prosser, and M. J. Johnson, "Stereo 3d tracking of infants in natural play conditions," in *Rehabilitation Robotics (ICORR), 2017 International Conference on*, IEEE, 2017, pp. 841–846.

[92] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[93] MATLAB, *version 79.3.0.713579 (R2017b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[94] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1605.06211*, 2016.

[95] M. Sun, P. Kohli, and J. Shotton, "Conditional regression forests for human pose estimation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 3394–3401.

[96] C. Sutton, A. McCallum, *et al.*, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[97] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[98] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[99] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[100] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[101] C. Garcia Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 2, pp. 577–584, Apr. 2017.

[102] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, IEEE, 1998, pp. 8–15.

[103] F. Wang and Y. Li, "Beyond physical connections: Tree models in human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 596–603.

[104] M. Sandau, H. Koblauch, T. B. Moeslund, H. Aanæs, T. Alkjær, and E. B. Simonsen, "Markerless motion capture can provide reliable 3d gait kinematics in the sagittal and frontal plane," *Medical engineering & physics*, vol. 36, no. 9, pp. 1168–1175, 2014.

[105] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt, "Fast articulated motion tracking using a sums of gaussians body model," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 951–958.

[106] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru, "An adaptable system for rgb-d based human body detection and pose estimation," *Journal of visual communication and image representation*, vol. 25, no. 1, pp. 39–52, 2014.

[107] K Nishi and J Miura, "Generation of human depth images with body part labels for complex human pose recognition," *Pattern Recognition*, vol. 71, pp. 402–413, 2017.

[108] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *International Conference on Computer Vision*, vol. 1, 2017, p. 5.

[109] H. Yub Jung, S. Lee, Y. Seok Heo, and I. Dong Yun, "Random tree walk toward instantaneous 3d human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2467–2474.

[110] M. Ding and G. Fan, "Generalized sum of gaussians for real-time human pose tracking from a single depth sensor," in *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2015, pp. 47–54.

[111] H. Mousavi Hondori and M. Khademi, "A review on technical and clinical impact of microsoft kinect on physical therapy and rehabilitation," *Journal of medical engineering*, vol. 2014, 2014.

[112] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley mhad: A comprehensive multimodal human action database," in *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, IEEE, 2013, pp. 53–60.

[113] C. Ionescu, F. Li, and C. Sminchisescu, "Latent structured models for human pose estimation," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2220–2227.

[114] I. Kostrikov and J. Gall, "Depth sweep regression forests for estimating 3d human pose from images.," in *BMVC*, vol. 1, 2014, p. 5.

[115] H. Y. Jung, S. Lee, Y. S. Heo, I. D. Yun, U Hankuk, and U Soonchunghyang, "Random tree walk toward instantaneous 3d human pose estimation,"

[116] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real-time human pose tracking from range data," 2012.

[117] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall, "A dual-source approach for 3d pose estimation from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4948–4956.

[118] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 1263–1272.

[119] G. Rogez and C. Schmid, "Image-based synthesis for deep 3d human pose estimation," *International Journal of Computer Vision*, pp. 1–16, 2018.

[120] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3d human pose estimation," *arXiv preprint arXiv:1804.01110*, 2018.